

IMPLEMENTACION DE INGENIERIA CONCEPTUAL, BASICA Y DE DETALLE  
PARA EL DISEÑO DE UN SISTEMA ELECTRONICO DE MONITOREO DE  
EQUIPOS DE APOYO DE AIRE ACONDICIONADO DE LA CENTRAL  
TELEFÓNICA DE COLON

CRISTIAN FELIPE BRAVO  
DANIEL FELIPE CAICEDO MEDINA

UNIVERSIDAD AUTONOMA DE OCCIDENTE  
FACULTAD DE INGENIERIA  
DEPARTAMENTO DE AUTOMATICA Y ELECTRONICA  
PROGRAMA DE INGENIERIA ELECTRONICA  
SANTIAGO DE CALI  
2007

IMPLEMENTACION DE INGENIERIA CONCEPTUAL, BASICA Y DE DETALLE  
PARA EL DISEÑO DE UN SISTEMA ELECTRONICO DE MONITOREO DE  
EQUIPOS DE APOYO DE AIRE ACONDICIONADO DE LA CENTRAL  
TELEFÓNICA DE COLON

CRISTIAN FELIPE BRAVO  
DANIEL FELIPE CAICEDO MEDINA

Pasantía para optar al título de Ingeniero Electrónico

Director  
OSCAR FERNANDO AGREDO  
Ingeniero Electrónico

UNIVERSIDAD AUTONOMA DE OCCIDENTE  
FACULTAD DE INGENIERIA  
DEPARTAMENTO DE AUTOMATICA Y ELECTRONICA  
PROGRAMA DE INGENIERIA ELECTRONICA  
SANTIAGO DE CALI  
2007

Nota de aceptación:

Aprobado por el Comité de Grado en cumplimiento de los requisitos exigidos por la Universidad Autónoma de Occidente para optar al título de Ingeniero Electrónico.

Ing. OSCAR FERNANDO AGREDO

---

Director

Santiago de Cali, 10 de Diciembre de 2007

## AGRADECIMIENTOS

A nuestros compañeros y amigos por la compañía durante este largo camino, a nuestros profesores y mentores por compartir su conocimiento pacientemente y principalmente a nuestros padres y hermanos por el apoyo incondicional, sin el cual esta meta nunca podría haber sido alcanzada.

## CONTENIDO

	Pág.
GLOSARIO	15
RESUMEN	17
INTRODUCCIÓN	18
1. FASE DE INGENIERIA CONCEPTUAL	19
1.1 PLANTEAMIENTO DEL PROBLEMA	19
1.2 MARCO TEORICO	19
1.2.1 Instrumentación	19
1.2.2 Comunicación	21
1.2.3 Interfaz en programación de software	21
1.2.4 Sistemas de control distribuidos (DCS)	25
1.2.5 Protocolos industriales (BITBUS)	28
1.2.6 Normas	33
1.2.7. Identificación	40
1.2.8 Norma UL	41
1.3 ANTECEDENTES	41
1.3.1 Optimización del monitoreo y control de la información analítica de la planta de acueducto del río cauca, desarrollado por Alexander Quijano zorrilla y Guillermo león González Córdoba año 2006	41
1.3.2 Sistema de detección de bloque de movimiento de rotación para el tornillo recolector, desarrollado por Ernesto López año 2006	42
1.3.3 Diseño e implementación de una red para control y monitoreo remoto de una planta de nivel, desarrollado por Carlos Eduardo Velásquez año 2003	43
1.3.4. Datalogger	43

1.3.5. 3710 ACM	44
1.4 OBJETIVO GENERAL	44
1.5 OBJETIVOS ESPEIFICOS	44
1.6 JUSTIFICACION	45
1.7 METODOLOGIA	45
1.8 CRONOGRAMA	47
1.9 PRESUPUESTO	48
1.10 FINANCIACION	48
2. INGENIERÍA BÁSICA: PRIMER PROTOTIPO	49
2.1 Adquisición de now how	49
2.1.1 Los microcontroladores	49
2.1.2 Comunicaciones	50
2.1.3 Comunicaciones nivel físico	59
2.1.4 Alimentación del sistema	61
2.1.5 Adaptación electrónica de señales análogas, digitales y manejo de tierras	62
2.1.6 Compilador de C para microcontroladores PIC	62
2.1.7 Cableado estructurado	62
2.2 Diagramas	63
2.3 Selección de normas a seguir	74
2.3.1 Simbología isa y sama	74
2.3.2 Norma UI	74
2.3.3 Norma retie	65

2.3.4 Norma IP	75
2.3.5 Estándar IEEE 802	75
2.4 Especificación de instrumentos y equipo de control	78
2.4.1 Tecnología microcontrolada	78
2.4.2 Conversores RS-232 /Ethernet TCP/IP	78
2.5 Especificaciones de montaje	81
2.5.1 Ubicación del equipo	81
2.6 Diagrama localización instrumentos	84
2.7 Listado y cotización de instrumentos y materiales	85
3. INGENIERÍA DE DETALLE: PROTOTIPO PROTOBOARD	86
3.1 Diagramas secuenciales, diagramas de flujo de variables de comunicaciones	86
3.1.1 Comunicaciones	86
3.2 .Tabla de entradas y salidas	117
3.3 Diseño de conexión entre señales de instrumentación de campo de la tarjeta	118
3.4 Diseño de tarjeta de comunicaciones y adquisición de datos	119
3.5 Cuantificación de obras para montajes	120
3.6 Plan de entrenamiento de personal	121
4. REALIZAR EVALUACION DISPOSITIVO FINAL	122
4.1 Prueba del prototipo en línea con PC	122
4.2 Prueba y puesta en marcha de comunicaciones	123
4.2.1 Comunicaciones del Sistema	123
4.2.2 Serial ASCII al PC	123

4.2.3 Serial ASCII del Sistema con las otras Comunicaciones	123
4.2.4 Comunicaciones del PC	125
4.3 Prueba y puesta en marcha de las estaciones de operación (software visual en PC). Ver anexo 4 manual de usuario	128
4.3.1 Generación de planos finales eléctricos	129
4.4 Configuración, arquitectura y comunicaciones sistemas de control	130
4.4.1 Diagrama del proceso de adquisición de datos	131
4.5 Especificaciones técnicas para montajes	131
5.CONCLUSIONES	132
BIBLIOGRAFIA	134
ANEXOS	136



## LISTA DE TABLAS

	Pág.
Tabla 1. Cronograma de actividades	47
Tabla 2. Presupuesto	48
Tabla 3. Comparación entre familias de microcontroladores	49
Tabla 4. Trama del mensaje RTU	52
Tabla 5. Transmisión de caracteres	53
Tabla 6. Set de instrucciones más utilizadas	54
Tabla 7. Lectura del estado espiral	55
Tabla 8. Lectura de registros 4x	55
Tabla 9. Lectura de registros 3x	56
Tabla 10. Lectura de estado ON/OFF	56
Tabla 11. Valor específico a registros	56
Tabla 12. Modificación de varios registros	57
Tabla 13. Especificaciones RS-485	61
Tabla 14. Versiones de 802.3	76
Tabla 15. Listado de materiales	85
Tabla 16. Entradas y salidas	117

## LISTA DE FIGURAS

	Pág.
Figura 1. Motor trifásico	20
Figura 2. Contactor	20
Figura 3. Lógica cableada	22
Figura 4. Esquema general	23
Figura 5. Esquema de programación	25
Figura 6. Arquitectura de un DCS	26
Figura 7. Datalogger	43
Figura 8 Diagrama de bloques del sistema en general	64
Figura 9. Diagrama de bloques del modbus rtu maestro	65
Figura 10. Diagrama de bloques del sistema señal ASCII	66
Figura 11. Diagrama de bloques del sistema que obtiene datos analógicos y digitales	67
Figura 12. Diagrama del sistema de calculo CRC	68
Figura 13. Diagrama del sistema de trasmisor paralelo recepción	69
Figura 14. Diagrama del sistema de trasmisor paralelo transmisión	70
Figura 15. Diagrama del software principal	71
Figura 16. Arquitectura de software	71
Figura 17. Diagrama de funcionamiento de la sección de Configuración	72
Figura 18 Diagrama de funcionamiento de la sección de Visor status	73

Figura 19. Diagrama de funcionamiento de la sección de Monitor	74
Figura 20. Microcontrolador PIC16F87A	78
Figura 21. Adaptador Ethernet a RS-232	79
Figura 22. Ubicación del equipo	82
Figura 23. Conexión de una sola tarjeta al PC	83
Figura 24. Plano de EMCALI ubicación instrumentos	84
Figura 25. Comunicación bidireccional	86
Figura 26. Comunicación bidireccional a 4 bits	87
Figura 27. Función Modbus	88
Figura 28. Identificador de función modbus	90
Figura 29. Generador de CRC	91
Figura 30. Transmisor de datos	91
Figura 31. Recepción de datos	92
Figura 32. Reconocimiento de información proveniente de la red	92
Figura 33. Código principal	94
Figura 34. Protocolo ASCII	95
Figura 35. Recepción de datos seriales	96
Figura 36. Código principal	97
Figura 37. Configuración del microcontrolador	98
Figura 38. Inicialización de variables, puertos, etc.	98
Figura 39. Sistema de transmisión de datos ASCII	99
Figura 40. Sistema de transmision de datos modbus rtu	100

Figura 41. Sistema de recepción de datos paralelo	100
Figura 42. Recepción de datos análogos	101
Figura 43. Recepción de datos digitales	102
Figura 44. Transmisión de datos al PC	103
Figura 45. Aceptación de información	103
Figura 46. Recoge datos	104
Figura 47. Cancela información	105
Figura 48. Ethernet o serial	105
Figura 49. Trama Modbus a enviar	106
Figura 50. Función carga	107
Figura 51. Función para cambiar ventana	107
Figura 52. Función reconocedora de datos serial RS-232	108
Figura 53. Función reconocedora de datos udp	110
Figura 54. Función salir	112
Figura 55. Función timer	112
Figura 56. Load configuración inicial	113
Figura 57. Recolección de datos udp	114
Figura 58. Diagrama de entradas y salidas	117
Figura 59. Subsistemas	118
Figura 60. Características físicas	119
Figura 61. Conexión	119
Figura 62. Simulación híper terminal	122
Figura 63. Trama de datos ASCII	123

Figura 64. Trama de datos ASCII a Modbus rtu	124
Figura 65. Respuesta de la trama de datos	125
Figura 66. Reenvío de información	125
Figura 67. Ventana de interfaz VB	126
Figura 68. Configuración interfaz VB	127
Figura 69. Datos de la red	128
Figura 70. Prueba de datos	128
Figura 71. Dispositivo caja negra	129
Figura 72. Diagrama eléctrico simulación	129
Figura 73. Arquitectura DCS	130
Figura 74. Diagrama del proceso de adquisición de datos	131

## LISTA DE ANEXOS

	Pág.
Anexo A. Tipos de medios de transmisión	136
Anexo B. Selección de normas a seguir	141
Anexo C. Hoja de especificaciones	147
Anexo D. Manual de operación y mantenimiento	148

## GLOSARIO

**BASE DATOS:** es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos tienen formato electrónico, que ofrece un amplio rango de soluciones al problema de almacenar datos.

**DCE:** equipo de comunicación de datos o equipo de terminación de datos. Los dispositivos y las conexiones de una red de comunicaciones que comprenden el extremo de la red de la interfaz de red del usuario. Proporciona una conexión física a la red, envía tráfico y proporciona una señal de sincronización utilizada para sincronizar la transmisión de datos entre los dispositivos DCE y DTE. Los modems y tarjetas de interfaz son ejemplos de DCE.

**DRIVER:** controlador de dispositivo, es un programa informático que permite al sistema operativo interactuar con un periférico, haciendo una abstracción del hardware y proporcionando una interfaz para usarlo. Se puede esquematizar como un manual de instrucciones que le indica cómo debe controlar y comunicarse con un dispositivo en particular. Por tanto, es una pieza esencial, sin la cual no se podría usar el hardware.

**DTE:** dispositivo en el extremo del usuario de una interfaz de red de usuario que sirve como origen de datos, destino de datos o ambos. Se conecta a una red de datos a través de un dispositivo DCE y utiliza señales de sincronización generadas por el DCE. DTE incluye dispositivos como computadoras, routers y multiplexores.

**ETHERNET:** especificación LAN de banda base inventada por Xerox Corporation y desarrollada conjuntamente por Intel y Digital Equipment Corporation. Las redes utilizan CSMA / CD y funcionan en distintos tipos de cable a 10, 100 y 1000 Mbps. Es similar a la serie de normas IEEE 802.3.

**INTERFAZ:** una conexión entre dos dispositivos o sistemas. En la tecnología de enrutamiento una conexión de red.

**PUERTO:** una interfaz en un dispositivo de red. Un conector hembra en un patch panel que acepta un conector del mismo tamaño.

VISUAL BASIC: lenguaje de programación desarrollado por Alan Cooper para Microsoft. El lenguaje de programación es un dialecto de BASIC, con importantes añadidos. Su primera versión fue presentada en 1991 con la intención de simplificar la programación utilizando un ambiente de desarrollo completamente gráfico que facilitara la creación de interfaces gráficas y en cierta medida también la programación misma.



## RESUMEN

Para la elaboración del proyecto se realizó un estudio, a cerca de la necesidad de monitorear equipos de apoyo como aires acondicionados, banco de baterías, banco de rectificadores, instalados en la planta telefónica de Emcali ubicada en colon, cuyo requisito fue el poder identificar fallas que se presentan en sus sistemas principales y que afectan los modos de control y automatización para la correcta operación de la central de conmutación, que es la encargada de prestar el servicio de comunicaciones. Para cumplir con dicho objetivo fue necesario detallar la operación y el funcionamiento de los equipos, desarrollar una etapa de investigación con el fin de obtener resultados satisfactorios que permitieran generar una solución eficiente y así determinar cual alternativa es la mas indicada como propuesta de proyecto a implementar en la empresa.

## INTRODUCCIÓN

Debido a la necesidad del hombre de ejercer control sobre las diferentes variables o señales que lo rodean se vio obligado a generar recursos o alternativas que le permitieran hacer dicho control y tener mayor conocimiento acerca de los sistemas en los cuales el esta involucrado.

Parte de este problema es lograr la solución de la información adquirida, que pueda ser monitoreada, gestionada desde un solo lugar y con la ayuda de herramientas tener control eficiente de los procesos que entregan dichas variables.

A raíz de los grandes avances tecnológicos en el sector de las comunicaciones, las empresas tomaron conciencia en la necesidad de actualizarse de tal forma, ser cada día más competitivos y hacer un uso eficiente y eficaz de todos sus equipos que conforman su infraestructura.

Con base en lo anterior, la finalidad de este proyecto es brindar una solución en la gestión y el monitoreo de equipos remotos en la empresa, buscando minimizar los costos de operación y la optimización del tiempo.

En terminación es diseñar un dispositivo capaz de monitorear estados de operación a través de una interfaz con un PC y permitir al usuario u operario poder almacenar datos estadísticos para luego ser analizados y presentar informes que ayuden a mejorar la eficiencia de todos los sistemas en la empresa.

## 1. INGENIERIA CONCEPTUAL.

### 1.1. PLANTEAMIENTO DEL PROBLEMA

La central de conmutación telefónica de EMCALI ubicada en colon, dispone de equipos de apoyo remotos como son aires acondicionados, bancos de baterías y bancos de rectificadores para el correcto funcionamiento, del servicio básico de telefonía en la ciudad.

Estos equipos proporcionan otro índice de seguridad en la central telefónica de ahí su importancia, a partir de esta idea surge la siguiente cuestión, ¿Qué problemas traería el que no halla una continua supervisión en tiempo real de los equipos de apoyo?, una pregunta importante, debido a que estos equipos son de vital importancia, si no se conoce su estado de operación, trae consigo diversos problemas. Por ejemplo, si estos están operando, por debajo de los límites o por encima, podría generar daños en los equipos de operación, lo cual se reflejaría en la calidad del servicio prestado. También se vería limitado con el mantenimiento de los equipos porque se tendría conocimiento de que el sistema no funciona cuando el daño ya esta hecho, eso significaría que no podría hacerse mantenimiento preventivo, lo cual elevaría los gastos de mantenimiento que es muy perjudicial para la empresa.

### 1.2. MARCO TEORICO

Este proyecto involucra las áreas de instrumentación, comunicación y programación de software. Por tal razón a continuación se realiza una presentación de los aspectos más relevantes de cada uno de ellas relacionados con el proyecto.

1.2.1. Instrumentación. La instrumentación y la teoría de control basan sus desarrollos en la necesidad de adquirir señales que provienen del medio con el fin de ser procesadas y analizadas.

Tener presente que toda instrumentación comienza con el sensor, un buen conocimiento de éstos traerá como consecuencia proyectos seguros, óptimos y rentables.

✓ Detección de Fases (Trifásicos). Aunque el conocer el estado de los contactos que entran a trabajar es importante, no me garantiza que el

dispositivo entre a funcionar, es por eso la importancia de conocer el funcionamiento de las fases.

Figura 1. Motor trifásico



Fuente: Bombas Centrifugas [en línea]. Buenos Aires: Climatecnica, 2005. [Consultado 3 de noviembre. de 2007]. Disponible en Internet: <http://www.climatecnica.com/bombas-centrifugas-horizontales-grundfos-ch-2-y-ch-4.72.360.html>

En la figura 1 se muestra un motor trifásico muy común en procesos industriales.

✓ Contactos electromagnéticos. Son sistemas ON/OFF los cuales se utilizan para poder hacer esquemas de mando y control, en los esquemas de potencia, se usan mucho para indicar alarmas.

Figura 2. Contactor



Fuente: Circuit Breaker [en línea]. Miami: Lake Wood, 2007. [Consultado 15 de julio. de 2007]. Disponible en Internet: [http://www.lakewoodconferences.com/catalog/87/437/circuit\\_breaker.html](http://www.lakewoodconferences.com/catalog/87/437/circuit_breaker.html)

En la figura 2 se muestra la gran variedad de estos dispositivos en el mercado, que son producidos por Siemens, telemecanique, general, etc.

1.2.2. Comunicación. Las comunicaciones se han convertido en una herramienta primordial en el normal desarrollo de los sistemas, se ha vuelto una necesidad poder administrar la información de forma clara, rápida y fácil de manipular, a partir de esto es que han surgido diversos métodos de comunicación que permiten el enlace entre un punto a otro, o entre varios puntos, se mencionaran los más usados.

✓ RS-232. Es una interfaz (también conocido como EIA RS-232C) que designa una norma para el intercambio serie de datos binarios entre un DTE (Equipo Terminal de datos) y un DCE (Equipo de terminación del circuito de datos) El RS-232 consiste en un conector tipo DB-25 de 25 pines, aunque es normal encontrar la versión de 9 pines DB-9, más barato e incluso más extendido para cierto tipo de periféricos (como el ratón serie del PC).

✓ RS-485 Transmisión balanceada diferencial. Cuando se requieren mayores distancias y velocidades de transmisión, entonces deben de emplearse las normas RS422 y RS485. Además, estas normas permiten también la transmisión multipunto, es decir una computadora central conectada con varias UTR. Dado que la computadora central típicamente tiene como salida la interfaz RS232, se hace necesaria la conexión de un módulo convertidor RS232 a RS422/485, objeto del presente proyecto, para implementar una red.

La transmisión diferencial permite velocidades de hasta 10 Mbps, sobre distancias de hasta 1.3 Km. Se usan dos señales para transmitir y dos para recibir, además de la tierra, la cual es normalmente conectada al blindaje del cable. En cada par, viajan la señal de transmisión y su complemento. En el receptor, la señal original se obtiene restando una de la otra. Esta técnica reduce grandemente el ruido generado en la línea, éste se induce por igual en ambas líneas del par y es al final cancelado. Este tipo de transmisión debe de hacerse siempre sobre cable del tipo "par trenzado" (twisted pairs).

1.2.3. Interfaz en programación de software. Visual Basic Es un lenguaje de fácil aprendizaje pensado tanto para programadores principiantes como expertos, guiado por eventos, y centrado en un motor de formularios poderoso que facilita el rápido desarrollo de aplicaciones gráficas. Su principal innovación, que luego fue adoptada por otros lenguajes, fue el uso de un tipo de DLL, llamado inicialmente VBX y posteriormente OCX, que permiten contener toda la funcionalidad de un control y facilitar su rápida incorporación a los formularios.

Su sintaxis, derivada del antiguo BASIC, ha sido ampliada con el tiempo al agregarse las características típicas de los lenguajes estructurados modernos.

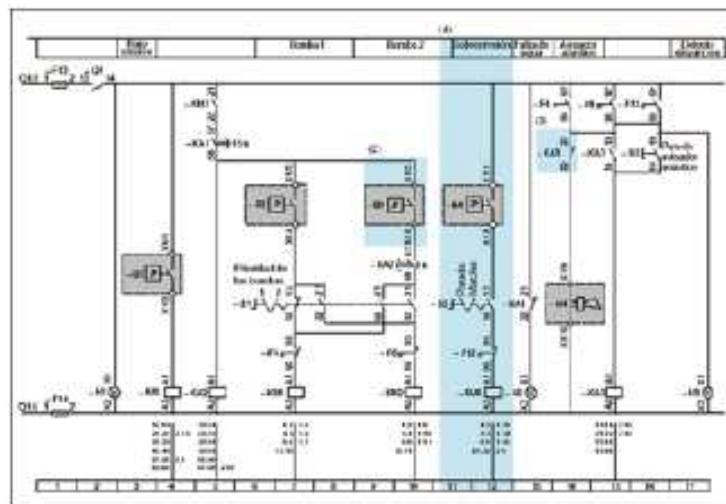
Se ha agregado una implementación limitada de la Programación Orientada a Objetos (los propios formularios y controles son objetos), aunque si que admite el polimorfismo mediante el uso de los Interfaces no admite la herencia. No requiere de manejo de punteros y posee un manejo muy sencillo de cadenas de caracteres. Posee varias bibliotecas para manejo de bases de datos, pudiendo conectar con cualquier base de datos a través de ODBC (Informix, DBase, Access, MySQL, SQL Server, PostgreSQL, etc.) a través de ADO.

Es utilizado principalmente para aplicaciones de gestión de empresas, debido a la rapidez con la que puede hacerse un programa que utilice una base de datos sencilla, además de la abundancia de programadores en este lenguaje.

El compilador de Microsoft genera ejecutables que requieren una DLL para que sus ejecutables funcionen, en algunos casos llamada MSVBVMxy.DLL (acrónimo de "Microsoft Visual Basic Virtual Machine x.y", siendo x.y la versión) y en otros VBRUNXXX.DLL ("Visual Basic Runtime X.XX"), que provee todas las funciones implementadas en el lenguaje. Además existen un gran número de bibliotecas (DLL) que facilitan el acceso a muchas funciones del sistema operativo y la integración con otras aplicaciones.

✓ Lógica cableada. La lógica cableada es una forma de realizar controles, en la que el tratamiento de datos (botonería, fines de carrera, sensores, presostatos, etc.), se efectúa en conjunto con contactores o relés auxiliares, frecuentemente asociados a temporizadores y contadores.

Figura 3. Lógica cableada

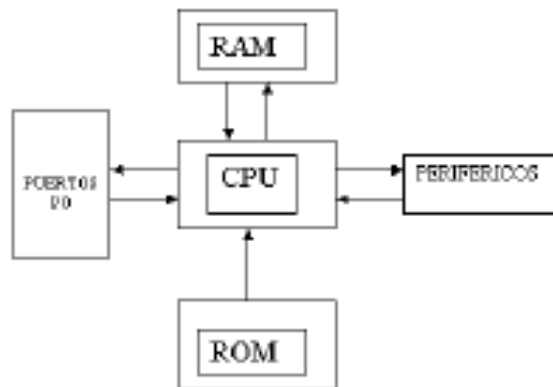


En la figura 3 se muestra un esquema general de lógica cableada.

A través de la conexión de los contactos de los diferentes elementos involucrados, se ejecutan secuencias de activación, desactivación y temporizaciones de los diferentes elementos que permiten realizar el manejo de la maquinaria (contactores, válvulas, pistones, calefactores, motores, etc.). El uso de relés auxiliares hace posible incrementar la cantidad de contactos disponibles para realizar la lógica, lo cual habitualmente es necesario, además de servir de interfaz al manejar diferentes niveles de voltaje (24 a 220 Vac y viceversa, por ejemplo). Las temporizaciones también son recurrentes, por lo cual uno o más temporizadores son comúnmente encontrados en estos sistemas.

✓ Microcontroladores. Un microcontrolador es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada y salida y periféricos. Estas partes están interconectadas dentro del microcontrolador, y en conjunto forman lo que se le conoce como microcomputadora. Se puede decir con toda propiedad que un microcontrolador es una microcomputadora completa encapsulada en un circuito integrado.

Figura 4. Esquema general



Toda microcomputadora requiere de un programa para que realice una función específica. La figura 4 muestra un esquema general. Este se almacena normalmente en la memoria ROM. No está de más mencionar que sin un programa, los microcontroladores carecen de utilidad.

El propósito fundamental de los microcontroladores es el de leer y ejecutar los programas que el usuario le escribe, es por esto que la programación es una actividad básica e indispensable cuando se diseñan circuitos y sistemas que los incluyan. El carácter programable de los microcontroladores simplifica el diseño de circuitos electrónicos. Permiten modularidad y flexibilidad, un mismo

circuito se puede utilizar para que realice diferentes funciones con solo cambiar el programa del microcontrolador.

Las aplicaciones de los microcontroladores son vastas, se puede decir que solo están limitadas por la imaginación del usuario. Es común encontrar microcontroladores en campos como la robótica y el automatismo, en la industria del entretenimiento, en las telecomunicaciones, en la instrumentación, en el hogar, en la industria automotriz, etc.

✓ Programación en microcontroladores. Los microcontroladores están diseñados para interpretar y procesar datos e instrucciones en forma binaria. Patrones de 1's y 0's conforman el lenguaje máquina de los microcontroladores, y es lo único que son capaces de entender. Estos 1's y 0's representan la unidad mínima de información, conocida como bit, solo puede adoptar uno de dos valores posibles: 0 ó 1.

La representación de datos, instrucciones y señales en forma de bits resulta dificultosa y tediosa para aquellas personas que no estén familiarizadas con el sistema de numeración binario. Aún para los usuarios expertos no resulta tan evidente la interpretación de instrucciones en forma binaria o lenguaje máquina (el lenguaje máquina se le conoce también como lenguaje de bajo nivel debido a que las instrucciones no son propias del lenguaje humano). Es por esto que la programación comúnmente se lleva a cabo en un lenguaje de alto nivel, es decir, un lenguaje que utilice frases o palabras semejantes o propias del lenguaje humano. Las sentencias de los lenguajes de alto nivel facilitan enormemente la programación, son familiares a nuestra manera de comunicarnos. Lenguajes como el C o BASIC son comúnmente utilizados en la programación de microcontroladores.

Otro tipo de lenguaje mas especializado es el lenguaje ensamblador. El lenguaje ensamblador es una lista con un limitado numero instrucciones a los cuales puede responder un microcontrolador. Estas instrucciones son palabras o abreviaciones que representan las instrucciones en lenguaje máquina del microcontrolador.

Las instrucciones en lenguaje ensamblador, también conocidas como nemotécnicos, son fáciles de entender y permiten operar directamente con los registros de memoria así como con las instrucciones intrínsecas del microcontrolador. Es por esto que el lenguaje ensamblador es sin lugar a dudas el lenguaje por excelencia en la programación de microcontroladores, porque permite hacer un uso eficiente de la memoria y minimizar el tiempo de ejecución de un programa. Cualquiera que sea el lenguaje que se utilice en la programación de microcontroladores, es de lo mas recomendable profundizar

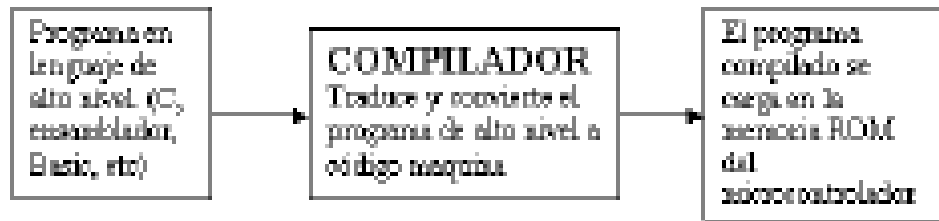


en su arquitectura interna, con este conocimiento se pueden aprovechar mas y mejor las capacidades de un microcontrolador dado.

Todo programa escrito en un lenguaje de alto nivel debe ser transformado en código maquina. Los programas que escribimos los entendemos nosotros, no así el microcontrolador. Un software de computadora, llamado compilador, traduce y transforma nuestro programa en código maquina, que es lo que realmente puede leer e interpretar el microcontrolador.

Una vez compilado el programa, es momento de transferir nuestro código maquina hacia la memoria interna del microcontrolador, usualmente hacia la ROM. Para esta tarea se utiliza un programador físico, que es una pieza de hardware que tiene el propósito de escribir el programa en la memoria interna del micro.

Figura 5. Esquema de programación



La figura 5 muestra un esquema general de programación en microcontroladores.

1.2.4. Sistemas de control distribuidos (DCS). El estudio de la arquitectura de un DCS se basa en el concepto de la pirámide de automatización o estructuración jerárquica de categorías de control en bloques o sectores de sistemas con funciones específicas que se muestra en la figura. También denominado Pirámide de Control.

Figura 6. Arquitectura de un DCS



En el nivel de campo o reflexivo se encuentran los sensores, los elementos finales de control o actuadores, los controladores reflexivos ver figura 6 (como válvulas reguladoras de presión) y los dispositivos de seguridad (válvulas de alivio, switches, etc.). A este nivel le corresponde el primer contacto de la tecnología con el proceso para medir, observar o monitorear, o para reflejar la acción final de una estrategia de control.

En el nivel de control directo o reactivo la información de campo es procesada por medio de una estrategia de control (combinatorio, secuencial, PID, FUZZY, etc.) por los denominados controladores de proceso como son los PLC o autómatas programables, los controladores inteligentes de proceso o de lazo (IPC), las unidades de acceso remoto (denominadas inicialmente como RTU y actualmente como I/O Remotas) y los procesadores de control específicos (variadores de velocidad, controladores de posición, etc.)

En el nivel de supervisión del control se tiene la visión del proceso para la coordinación de operaciones de maniobra y mantenimiento, la optimización de recursos (materias primas, energía, etc.) la generación de procesos estadísticos representados en la información de bases de datos. Los dispositivos de este nivel son computadores o paneles de operación donde se encuentran las estaciones de operación o HMI, las bases de datos para variables de control y los habilitadores de aplicación (AE). Estos recursos deben estar en capacidad de implementar: la diagramación del proceso en forma de planos y copias de seguridad, el monitoreo o visualización de las variables y sus tendencias, el mando (comando) coordinado y pertinente para cambio de consignas y parámetros, la generación de reportes o protocolos de impresión de variables, avisos y alarmas, el enlace como servidor de datos a otras aplicaciones (AE) del mismo nivel o de niveles superiores.

El nivel de gestión de producción se encarga de la administración y toma de decisiones que implican producción, costos, inventarios, manejo del mantenimiento preventivo y predictivo, la programación del producto final de la planta en cantidad, costo, entregas a tiempo y documentación. En este nivel se emplean software como MES (Sistemas de ejecución de manufactura), MRP (Planeación de requerimientos de material), CIM (Computadores integrados hacia la manufactura) y MP (paquetes de mantenimiento)

En el nivel de manejo de planta o de toma de decisiones corporativas generalmente esta asociado a la utilización de un conjunto de paquetes de software que se clasifican como MIS (Management Information System).

Con este concepto de pirámide de control los fabricantes de software y hardware industrial presentan soluciones en la medida de su tecnología y sector del mercado que atiendan. Algunos no tienen en cuenta el último nivel y lo integran al nivel de gestión producción. Otros fabricantes presentan sus pirámides de control desde el punto de vista de los enlaces de comunicaciones que brinda la tecnología actual.

✓ Comunicaciones en un (DCS). El concepto de comunicaciones en un DCS se basa en el modelo OSI de ISO. El modelo divide en siete capas o niveles el problema de las comunicaciones en la pirámide de control. La capa 1 es hardware y de la 2 a la 7 es software. En los DCS las comunicaciones son seriales y se dividen en redes (niveles de gerencia, gestión y supervisión) que implementan del modelo las capas 1, 2, 6 y 7 y Subredes o buses de campo (niveles de control y campo) que implementan del modelo las capas 1,2 y 7.

- La capa 1 soluciona la forma de conexión física de las estaciones del DCS (PC's, PLC's, HMI's, I/O's, tec). Generalmente los medios más usados son Ethernet, RS422, RS485, RS232, 20 MA o TTY, conexión fibra óptica, conexión inalámbrica RF (27, 200, 900, 2400 y 5200 Mhz) y conexiones especiales como 4.20MA (Fieldbus), 24v (As interfase), etc.

- La capa 2 se encarga de los enlaces punto a punto de estaciones y coordina el sincronismo de bit y el manejo de errores. El sincronismo de trama o enlaces multipunto y control de acceso al medio (MAC).

- Las capas 6 y 7 en los DCS tienen las funciones de presentación y formato de los datos de proceso a las aplicaciones en las estaciones.

✓ Software y algoritmos en un (DCS). El término software en este punto se refiere a los programas, lenguajes y normas que se utilizan para poder implementar los algoritmos de control en los controladores, los programas en las estaciones de operación o HMI, la configuración de los enlaces de comunicaciones, las áreas de memoria compartidas y las direcciones de red. El término algoritmos se refiere a las estrategias de la teoría de control implementadas en un DCS en forma digital y que en muchos casos están normalizadas en estándares de IEC, ISA o IEEE.

1.2.5. Protocolos industriales (BITBUS). Es un sistema de comunicación serie para uso industrial, normalmente denominado bus de campo. Está basado en una línea compartida RS-485 (varias estaciones de comunicación en un mismo par de cables) y está optimizado para la transmisión de pequeños mensajes en tiempo real. En instalaciones más actuales se emplea también fibra óptica para su implementación.

Una red de comunicación BITBUS siempre posee un maestro y uno o varios esclavos. Cada esclavo posee su propia dirección de red que le hace diferenciable e identificable dentro de la red. El maestro maneja la red seleccionando los esclavos. Los esclavos deben responder exclusivamente cuando son requeridos por el maestro. Este simple método de control permite una alta seguridad en la comunicación, optimizada para la mayoría de las aplicaciones de automatización.

La estructura de una red BITBUS siempre se compone de un maestro, un número de esclavos y, si es necesario, uno o varios repetidores. Los repetidores deben ser empleados si hay presentes más de 28 estaciones de trabajo a lo largo del segmento de cable que se va a emplear.

Cuando se emplea en ambientes con ruido industrial, es recomendable una separación galvánica de la interfase BITBUS: el RS-485 está ópticamente aislado del controlador. La utilización de un cable de fibra óptica también es posible, pero no obligatoria.

La experiencia muestra que un buen cable blindado, con una referencia a tierra, combinado con la señal diferencial del RS-485 es un medio físico excelente, incluso en ambientes ruidosos. Y además es barato.

Una característica importante de BITBUS es que está aceptado internacionalmente como un estándar industrial: enlaza equipos de diferentes fabricantes, siendo el hardware y el software totalmente estandarizados y compatibles. En 1991 el BITBUS fue oficialmente estandarizado como un

estándar internacional IEEE-1118. El estándar IEEE-1118 contiene la funcionalidad del clásico BITBUS de Intel como base e incorpora nuevas funciones.

Las implementaciones clásicas del protocolo BITBUS están basadas en el microcontrolador i8044: este microcontrolador contiene el soporte completo de las especificaciones BITBUS, incluyendo el controlador RAC y un pequeño núcleo operador en tiempo real. Las implementaciones modernas están basadas en diferentes procesadores y microcontroladores, que presentan un bajo consumo de potencia y un alto rendimiento de transferencia para BITBUS.

Una de las razones del éxito de BITBUS son los servicios RAC. El RAC (Remote Access Control) es una serie de servicios destinados a proporcionar un acceso directo a recursos remotos, tareas, E/S, memoria, etc. En el modelo OSI, las funciones RAC cubren la capa de aplicación (capa 7).

A pesar que BITBUS es un bus de campo algo antiguo, emplea un protocolo moderno: la comunicación está basada en el protocolo SDLC, inventado por IBM, que es empleado también en comunicaciones Ethernet e ISDN.

Características generales:

- ❖ Tipo: Red maestro-esclavo mediante tramas de mensaje. 248 bytes máx. de longitud por mensaje.
- ❖ Estructura: Bus, terminado en ambos extremos. Prolongación posible mediante el empleo de repetidores.
- ❖ Soporte: Par de cables trenzados (un par, impedancia característica de 120) con toma de tierra y apantallado. Segundo par necesario para operar con repetidor.
- ❖ Niveles eléctricos: Par diferencial 0/5V según está definido en RS-485.
- ❖ Protocolo: SDLC con NRZI sincronizado de reloj propio (auto reloj) con flags de apertura y cierre, testeo de direccionamiento y comprobación de palabra por CRC de 16 bits.
- ❖ Tasa de transferencia: 375k Bit/s o 62,5k Bit/s.

- ❖ Esclavos: 28 por segmento con repetidores después de cada segmento, 250 máximo. Tasa de transferencia con más de un repetidor: 62,5K Bit/s exclusivamente.

- ❖ Longitud: 300m por segmento a 375k Bit/s, 1200m a 62,5kBit/s.

- ❖ Conector: Conector Sub-D de 9 pines.

✓ (MODBUS). Es un protocolo de comunicaciones situado en el nivel 7 del Modelo OSI, basado en la arquitectura maestro/esclavo o cliente/servidor, diseñado en 1979 por Modicon para su gama de controladores lógicos programables (PLCs). Convertido en un protocolo de comunicaciones estándar de facto en la industria es el que goza de mayor disponibilidad para la conexión de dispositivos electrónicos industriales. Las razones por las cuales el uso de Modbus es superior a otros protocolos de comunicaciones son:

- ❖ es público
- ❖ su implementación es fácil y requiere poco desarrollo
- ❖ maneja bloques de datos sin suponer restricciones

Modbus permite el control de una red de dispositivos, por ejemplo un sistema de medida de temperatura y humedad, y comunicar los resultados a un ordenador. Modbus también se usa para la conexión de un ordenador de supervisión con una unidad remota (RTU) en sistemas de supervisión adquisición de datos (SCADA). Existen versiones del protocolo Modbus para puerto serie y Ethernet (Modbus/TCP).

Modbus RTU es una representación binaria compacta de los datos. Modbus ASCII es una representación legible del protocolo pero menos eficiente. Ambas implementaciones del protocolo son serie. El formato RTU finaliza la trama con un suma de control de redundancia cíclica (CRC), mientras que el formato ASCII utiliza una suma de control de redundancia longitudinal (LRC). La versión Modbus/TCP es muy semejante al formato RTU, pero estableciendo la transmisión mediante paquetes TCP/IP.

Modbus Plus (Modbus+ o MB+), es una versión extendida del protocolo que permanece propietaria de Modicon. Dada la naturaleza de la red precisa un coprocesador dedicado para el control de la misma. Con una velocidad de 1 Mbit/s en un par trenzado sus especificaciones son muy semejantes al estándar EIA/RS-485 aunque no guarda compatibilidad con este.

Cada dispositivo de la red Modbus posee una dirección única. Cualquier dispositivo puede enviar órdenes Modbus, aunque lo habitual es permitirlo sólo

a un dispositivo maestro. Cada comando Modbus contiene la dirección del dispositivo destinatario de la orden. Todos los dispositivos reciben la trama pero sólo el destinatario la ejecuta (salvo un modo especial denominado "Broadcast"). Cada uno de los mensajes incluye información redundante que asegura su integridad en la recepción. Los comandos básicos Modbus permiten controlar un dispositivo RTU para modificar el valor de alguno de sus registros o bien solicitar el contenido de dichos registros.

Existe gran cantidad de modems que aceptan el protocolo Modbus. Algunos están específicamente diseñados para funcionar con este protocolo. Existen implementaciones para conexión por cable, wireless, SMS o GPRS. La mayoría de problemas presentados hacen referencia a la latencia y a la sincronización. Todas las implementaciones presentan variaciones respecto al estándar oficial.

Algunas de las variaciones más habituales son:

- Tipos de Datos
  - Coma Flotante IEEE
  - entero 32 bits
  - datos 8 bits
  - tipos de datos mixtos
  - campos de bits en enteros
  - multiplicadores para cambio de datos a/de entero. 10, 100, 1000, 256 ...
- Extensiones del Protocolo
  - direcciones de esclavo de 16 bits
  - Tamaño de datos de 32 bits (1 dirección = 32 bits de datos devueltos.)

✓ (PROFIBUS). Profibus (Process Field Bus) es posiblemente el bus de campo industrial con mayor número de nodos instalados, en el año 2004 se calculaba un total de 12,6 millones de nodos. Se trata de una red abierta, estándar e independiente de cualquier fabricante, cuenta con varios perfiles y se adapta a las condiciones de las aplicaciones de automatización industrial.

Fue desarrollada en el año 1987 por las empresas alemanas Bosch, Klöckner Möller y Siemens. En 1989 la adoptó la norma alemana DIN19245 y fue confirmada como norma europea en 1996 como EN50170. En el año 2002 se actualizaron incluyendo la versión para Ethernet llamada Profinet.

Este tipo de red trabaja con nodos maestros y nodos esclavos. Los nodos maestros se llaman también activos y los esclavos pasivos.

Además, junto con las especificaciones de otros buses de campo se recoge en las normas internacionales IEC61158 e IEC61784.

Características:

- Velocidades de transmisión:

9.6, 19.2, 93.75, 187.5 y 500 KBaudios.

- Número máximo de estaciones: 127 (32 sin utilizar repetidores).
- Distancias máximas alcanzables (cable de 0.22 mm. de diámetro):

Hasta 93.75 KBaudios: 1200 metros 187.5 KBaudios: 600 metros 500 KBaudios: 200 metros

- Estaciones pueden ser activas (maestros) o pasivas (esclavos).
- Conexiones de tipo bidireccionales, multicast o broadcast.

✓ (ETHERNET). Es el nombre de una tecnología de redes de computadoras de área local (LANs) basada en tramas de datos. El nombre viene del concepto físico de ether. Ethernet define las características de cableado y señalización de nivel físico y los formatos de trama del nivel de enlace de datos del modelo OSI. Ethernet se refiere a las redes de área local y dispositivos bajo el estándar IEEE 802.3 que define el protocolo CSMA/CD, aunque actualmente se llama Ethernet a todas las redes cableadas que usen el formato de trama descrito más abajo, aunque no tenga CSMA/CD como método de acceso al medio.

Ethernet es la capa física más popular de la tecnología LAN usada actualmente, que fue desarrollada principalmente por la empresa XEROX. Otros tipos de LAN incluyen Token Ring 802.5, Fast Ethernet, FDDI, ATM y LocalTalk. Ethernet es popular porque permite un buen equilibrio entre velocidad, costo y facilidad de instalación. Estos puntos fuertes, combinados con la amplia aceptación en el mercado y la habilidad de soportar virtualmente todos los protocolos de red populares, hacen a Ethernet la tecnología ideal para la red de la mayoría de usuarios de la informática actual.

El estándar original IEEE 802.3 estuvo basado en la especificación Ethernet 1.0 y era muy similar. El documento preliminar fue aprobado en 1983 y fue publicado oficialmente en 1985 (ANSI/IEEE Std. 802.3-1985). Desde entonces un gran número de suplementos han sido publicados para tomar ventaja de los avances tecnológicos y poder utilizar distintos medios de transmisión, así como velocidades de transferencia más altas y controles de acceso a la red adicionales.

Los elementos en una red Ethernet son los nodos de red y el medio de interconexión. Dichos nodos de red se pueden clasificar en dos grandes grupos: Equipo Terminal de Datos (DTE) y Equipo de Comunicación de Datos (DCE). Los DTE son los dispositivos que generan o son el destino de los datos, tales como las computadoras personales, las estaciones de trabajo, los servidores de archivos, los servidores de impresión, todos son parte del grupo de estaciones finales. Mientras que los DCE son los dispositivos de red



intermediarios que reciben y retransmiten las tramas dentro de la red, y pueden ser ruteadores, conmutadores (switch), concentradores (HUB), repetidores, o interfaces de comunicación, como un módem o una tarjeta de interfase por ejemplo.

- NIC, o Tarjeta de Interfaz de Red - permite el acceso de una computadora a una red local. Cada adaptador posee una dirección MAC que la identifica en la red y es única. Una computadora conectada a una red se denomina nodo.
- Repetidor o repeater - aumenta el alcance de una conexión física, recibiendo las señales y retransmitiéndolas, para evitar su degradación a lo largo del medio de transmisión, lográndose un alcance mayor. Usualmente se usa para unir dos áreas locales de igual tecnología y sólo tiene dos puertos. Opera en la capa física del modelo OSI.
- Concentrador o *hub* - funciona como un repetidor, pero permite la interconexión de múltiples nodos, su funcionamiento es relativamente simple, recibe una trama de ethernet y la repite por todos sus puertos, sin llevar a cabo ningún proceso sobre las mismas. Opera en la capa física del modelo OSI.
- Puente o bridge - interconectan segmentos de red, haciendo el cambio de frames (tramas) entre las redes de acuerdo con una tabla de direcciones que dice en qué segmento está ubicada una dirección MAC.
- Conmutador o Switch - funciona como el bridge, pero permite la interconexión de múltiples segmentos de red, funciona en velocidades más rápidas y es más sofisticado. Los *switches* pueden tener otras funcionalidades, como *redes virtuales* y permiten su configuración a través de la propia red. Su funcionamiento básico es en las capas física y de enlace de datos del modelo OSI, por lo cual son capaces de procesar información de las tramas; siendo su funcionalidad más importante las tablas de dirección. Por ejemplo, una computadora conectada al puerto 1 del conmutador envía una trama a otra computadora conectada al puerto 2, el *switch* recibe la trama y la transmite a todos sus puertos, excepto aquel por donde la recibió, la computadora 2 recibirá el mensaje y eventualmente lo responderá, generando tráfico en el sentido contrario, por lo cual ahora el *switch* conocerá las direcciones MAC de las computadoras en el puerto 1 y 2, y cuando reciba otra trama con dirección de destino a alguna de ellas, sólo transmitirá la trama a dicho puerto, lo cual disminuye el tráfico de la red y contribuye al buen funcionamiento de la misma.

#### 1.2.6. Normas (Ver anexo 2)

✓ Simbología isa y sama. Para designar y representar los instrumentos de medición y control se emplean normas muy variadas que a veces varían de industria en industria. Esta gran variedad de normas y sistemas utilizados en las organizaciones industriales indica la necesidad universal de una normalización en este campo. Varias sociedades han dirigido sus esfuerzos en este sentido, y entre ellas se encuentra como una de las importantes la Sociedad de Instrumentos de Estados Unidos, ISA (Instrument Society of America) cuyas normas tienen por objeto establecer sistemas de designación (código y símbolos) de aplicación a las industrias químicas, petroquímicas, aire acondicionado, etc.

✓ Norma isa-s5.1

- Cada instrumento debe identificarse con sistema de letras que lo clasifique funcionalmente.

- El número de letras funcionales para un instrumento debe ser mínimo, no excediendo de cuatro. Para ello conviene:

- > Disponer las letras en subgrupos. Por ejemplo, un transmisor registrador de relación de caudales con un interruptor de alarma de relación de caudales puede identificarse con dos círculos uno con FFRT-3 y el otro FFS-3.

- > En un instrumento que indica y registra la misma variable medida puede omitirse la letra I (indicación).

- > Los bucles de instrumentos de un proyecto o secciones de un proyecto deben identificarse con una secuencia única de números. Esta puede empezar con el número 1 o cualquier otro número conveniente, tal como 301 o 1201 que puede incorporar información codificada tal como área de planta.

- > Si un bucle dado tiene más de un instrumento con la misma identificación funcional, es preferible añadir un sufijo, ejemplo FV-2A, FV-2B, FV-2C, etc., o TE-25-1, TE-25-2, TE-25-3, etc. Estos sufijos pueden añadirse obedeciendo a las siguientes reglas:

- Deben emplearse letras mayúsculas, A, B, C, etc.
- En un instrumento tal como un registrador de temperatura multipunto que imprime números para identificación de los puntos, los elementos primarios pueden numerarse TE-25-1, TE-25-2, TE-25-3, etc.

- Las subdivisiones interiores de un bucle pueden designarse por sufijos formados por letras y números.
  
- Un instrumento que realiza dos o más funciones puede designarse por todas sus funciones. Por ejemplo, un registrador de caudal FR-2 con pluma de presión PR-4 se designa preferentemente FR-2/PR-4 o bien UR-7; un registrador de presión de dos plumas como PR-7/8; y una ventanilla de alarma para temperatura alta y baja como TAH/L-9.
  
- Los accesorios para instrumentos tales como rotámetros de purga, filtros mano reductores y pots de sellos que no están representados explícitamente en un diagrama de flujo, pero que necesitan una identificación para otros usos deben tenerla de acuerdo con su función y deben emplear el mismo número del bucle que el del instrumento asociado. Alternativamente, los accesorios pueden emplear el mismo número de identificación que el de sus instrumentos asociados, pero con palabras aclaratorias si ello es necesario.
  
- > Por consiguiente, una brida para una placa-orificio FE-7 debe designarse como FX-7 o bien como FE-7 brida. Un rotámetro regulador de purga asociado con un manómetro PI-8 debe identificarse como FICV-8, pero puede también marcarse PI-8 purga. Una sonda empleada con un termómetro TI-9 será TW-9, o bien, TI-9 sonda.
  
- Para cubrir las designaciones no normalizadas que pueden emplearse repetidamente en un proyecto se han previsto letras libres. Estas letras pueden tener un significado como primera letra y otro, como letra sucesiva. Por ejemplo, la letra N puede representar como primera letra el módulo de elasticidad y como sucesiva un osciloscopio.
  
- La letra sin clasificar X, puede emplearse en las designaciones no indicadas que se utilicen sólo una vez o un número limitado de veces. Se recomienda que su significado figure en el exterior del círculo de identificación del instrumento. Ejemplo: XR-3 registrador de vibración.
  
- Cualquier letra primera si se utiliza con las letras de modificación D (diferencial), F (relación) o Q (integración) o cualquier combinación de las mismas cambia su significado para representar una nueva variable medida. Por ejemplo, los instrumentos TDI y TI miden dos variables distintas, la temperatura diferencial y la temperatura, respectivamente.

- La letra A para análisis, abarca todos los análisis no indicados en la tabla que no están cubiertos por una letra libre. Es conveniente definir el tipo de análisis al lado del símbolo en el diagrama de proceso.
- El empleo de la letra U como multivariable en lugar de una combinación de primeras letras, es opcional.
- El empleo de los términos de modificaciones alto, medio, bajo, medio o intermedio y exploración, es preferible pero opcional.
- El término seguridad, debe aplicarse sólo a elementos primarios y a elementos finales de control que protejan contra condiciones de emergencia (peligrosas para el equipo o el personal). Por esto motiva, una válvula autorreguladora de presión que regula la presión de salida de un sistema, mediante el alivio o escape de fluido al exterior, debe ser PCV, pero si esta misma válvula se emplea contra condiciones de emergencia, se designa PSV.
- > La designación PSV se aplica a todas las válvulas proyectadas para proteger contra condiciones de emergencia de presión sin tener en cuenta si las características de la válvula y la forma de trabajo la colocan en la categoría de válvula de seguridad, válvula de alivio, o válvula de seguridad de alivio.
- La letra de función pasiva vidrio, se aplica a los instrumentos que proporcionan una visión directa no calibrada del proceso.
- La letra indicación se refiere a la lectura de una medida real de proceso. No se aplica a la escala de ajuste manual de la variable si no hay indicación de ésta.
- Una luz piloto que es parte de un bucle de control debe designarse por una primera letra seguida de la letra sucesiva L. Por ejemplo, una luz piloto que indica un período de tiempo terminado se designará KL. Sin embargo, si se desea identificar una luz piloto fuera del bucle de control, la luz piloto puede designarse en la misma forma o bien alternativamente por una letra única L. Por ejemplo, una luz piloto de marcha de un motor eléctrico puede identificarse EL, suponiendo que la variable medida adecuada es la tensión, o bien XL, suponiendo que la luz es excitada por los contactos eléctricos auxiliares del arrancador del motor, o bien simplemente L. La actuación de la luz piloto puede ser acompañada por una señal audible.

- El empleo de la letra U como multifunción en lugar de una combinación de otras letras, es opcional.

- Se supone que las funciones asociadas con el uso de la letra sucesiva Y se definirán en el exterior del símbolo del instrumento cuando sea conveniente hacerlo así.

- Los términos alto, bajo y medio o intermedio deben corresponder a valores de la variable medida, no a los de la señal a menos que se indique de otro modo. Por ejemplo, una alarma de nivel alto derivada de una señal de un transmisor de nivel de acción inversa debe designarse LAH incluso aunque la alarma sea actuada cuando la señal cae a un valor bajo.

- Los términos alto y bajo, cuando se aplican a válvulas, o a otros dispositivos de cierre- apertura, se definen como sigue:

- Alto: indica que la válvula está, o se aproxima a la posición de apertura completa.

Bajo: denota que se acerca o está en la posición completamente cerrada.  
Figuran a continuación los símbolos a emplear en los planos y dibujos de representación de instrumentos en los procesos industriales.

Se sugieren las siguientes abreviaturas para representar el tipo de alimentación (o bien de purga de fluidos)

AB Alimentación de aire  
ES Alimentación eléctrica  
GS Alimentación de gas  
HS Alimentación hidráulica  
NS Alimentación de nitrógeno  
SS Alimentación de vapor  
WS Alimentación de agua

✓ Norma isa-s5.3. El objeto de esta norma es documentar los instrumentos formados por ordenadores, controladores programables, mini ordenadores y sistemas a microprocesador que disponen de control compartido, visualización compartida y otras características de interfase. Los símbolos representan la interfase con los equipos anteriores de la instrumentación de campo, de la instrumentación de la sala de control y de otros tipos de hardware.

El tamaño de los símbolos debe ser conforme a la norma ISA-S5.1, a la que complementa.

#### Símbolos de visualización del control distribuido/compartido

➤ Accesible normalmente al operador-indicador/controlador/registrator o punto de alarma.

- Visualización compartida.
- Visualización y control compartidos.
- Acceso limitado a la red de comunicaciones.
- Interfase del operador en la red de comunicaciones.
  
- Dispositivo de interfase auxiliar del operador.
  - Montado en panel; carátula analógica; no está montado normalmente en la consola principal del operador.
  - Controlador de reserva o estación manual.
  - El acceso puede estar limitado a la red de comunicaciones.
  - Interfase del operador vía la red de comunicaciones.
  
- No accesible normalmente al operador
  
- Controlador ciego compartido.
- Visualización compartida instalada en campo
  - Puede estar en la red de comunicaciones.
  - Normalmente operación ciega
  - Puede ser alterado por la configuración.

✓ Símbolos del ordenador. A utilizar cuando los sistemas incluyen componentes identificados como ordenadores, diferentes de un procesador integral que excita las varias funciones de un sistema de control distribuido. El componente ordenador puede ser integrado en el sistema, vía la red de datos, o puede ser un ordenador aislado.

➤ Normalmente accesible al operador-indicador/controlador/registrator o punto de alarma. Utilizado usualmente para indicar la pantalla de vídeo.

- Normalmente no accesible al operador.
- Interfase entrada/salida.
- Cálculo /acondicionamiento de señal dentro de un ordenador

- Puede usarse como un controlador ciego o como módulo de cálculo de software
- ✓ Símbolos de control lógico y secuencial.
- Símbolo general. Para complejos no definidos interconectando control lógico o secuencial.
- Control distribuido interconectando controladores lógicos con funciones lógicas binarias o secuenciales.
- Paquete de controlador lógico programable o controladores lógicos digitales integrados con el equipo de control distribuido.
- No accesible normalmente al operador.
- Control distribuido interconectando un controlador lógico con funciones lógicas binarias o secuenciales.
- Paquete de controlador lógico programable o controladores lógicos digitales integrados con el equipo de control distribuido.
- Accesible normalmente al operador.
- ✓ Símbolos de funciones internas del sistema
- Cálculo/acondicionamiento de señal.
- Para identificación de bloques consulte ISA-S5.
- Para requerimientos de cálculo amplios, use la designación <C>. Escriba aclaraciones en documentación suplementaria.
- Utilizado en combinación con válvulas de alivio según ISA-S5.1.
- ✓ Símbolos comunes.
- Red del sistema.

- Usado para indicar una red de software, o conexiones entre funciones suministradas en el sistema del fabricante.
- Alternativamente, la red puede ser mostrada implícitamente por símbolos contiguos.
- Puede utilizarse para indicar una red de comunicaciones a opción del usuario.

Registadores y otros sistemas de retención de datos históricos Los registradores convencionales, tales como los de gráfico de banda se mostrarán de acuerdo con ISA-S51, en los registradores asignables utilice el símbolo 1.

El almacenamiento en masa de largo plazo de una variable de proceso mediante memorias digitales como cinta, disco, etc., debe representarse de acuerdo con los símbolos de visualización de control distribuido/compartido o símbolos de ordenador de esta norma, dependiendo de la localización del aparato.

1.2.7. Identificación. Los códigos de identificación de esta norma deben cumplir con ISA-S5.1 con las siguientes adiciones.

✓ Alarmas de software. Las alarmas de software pueden ser identificadas situando letras de designación de la tabla 1.1 de ISA-S5.1 en las líneas de señal de entrada o de salida de los controladores, o de otro componente específico integral del sistema. Ver Alarmas que aparecen posteriormente.

✓ Símbolos. Pueden unirse dos o más símbolos para expresar los significados siguientes, además de los mostrados en ISA-S5.1:

- Comunicación entre los instrumentos asociados, por ejemplo, hilos de conexión, redes internas del sistema, reserva.
- Instrumentos integrados con funciones múltiples, por ejemplo, registrador multipunto, válvula de control con controlador incorporado.

La aplicación de símbolos contiguos es una opción del usuario. Si su aplicación no es absolutamente clara, los símbolos contiguos no deben utilizarse.



1.2.8. Norma ul. Underwriters Laboratories Inc. (UL) es un organismo independiente de pruebas ensayos de seguridad y certificación, que evalúa productos, materiales y sistemas trabajando por la seguridad de los consumidores y sus bienes desde 1894. UL es líder mundial en el desarrollo de normas de seguridad de productos, las cuales son frecuentemente actualizadas y revisadas para responder a los cambios en la tecnología o responder a nuevos usos de estos productos.

### 1.3. ANTECEDENTES

La naturaleza y complejidad de las actividades de automatización ha evolucionado y aumentado en función de la rapidez y amplitud de los progresos registrados en las tecnologías de la información y la comunicación.

En el valle diferentes universidades han planteado soluciones a problemáticas similares al planteado en EMCALI, cada una de estas me muestra una posible solución o un a idea nueva que puede ser utilizada como ayuda en nuestro proyecto.

1.3.1. Optimización del monitoreo y control de la información analítica de la planta de acueducto del río cauca, desarrollado por Alexander Quijano zorrilla y Guillermo león González Córdoba año 2006.

Actualmente la planta de acueducto río cauca no tiene un sistema que analice constantemente las variables químicas mas importantes del proceso de potabilización del agua por lo cual se hacen varias propuestas para implementar un sistema general de instrumentación que permita a los operadores un manejo mas efectivo y menos tedioso del proceso, se estudia 2 sistemas múltipara métricos con el objetivo de analizar por medio de requerimientos de usuario y técnicos cual es el mejor para implementar en el proyecto.

También se analizan sondas de PH, Turbiedad, oxígeno disuelta, un analizador de aluminio residual y un par de radios para realizar un enlace entre la planta puerto Mallarino y la planta río cauca.

Una vez que se definió la instrumentación de campo requerido se entro a seleccionar el sistema de comunicación que permitiera la centralización de la información y la jerarquización de los equipos. Un protocolo de bus de campo que a través de un enlace serial RS-485 permita conectar muchas estaciones

entre maestros y esclavos y maneje una velocidad rapsoda de transmisión de datos.

Con el uso de este sistema se requerirá la utilización de un maestro, quien seria el encargado de administrar el acceso al bus; para ello se planteo la utilización de un PLC maestro que aparte de cumplir esta función centralizara la información y fuera una interfase entre la red y un PC que se encargaría de mostrar los datos y recopilar la información de las variables a través de un sistema SCADA.

1.3.2. Sistema de detección de bloque de movimiento de rotación para el tornillo recolector, desarrollado por Ernesto López año 2006.

El proyecto esta compuesto de dos partes:

Modo de medición esta encargado de monitorear la velocidad del motor hidráulico que mueve el tronillo desarenador. Sus funciones son recibir las tramas diferentes provenientes del nodo configuración, calcular las RPM's, captura de pulsos provenientes de la detección del movimiento de rotación del tornillo desarenador, habilitación ON/OFF en hardware de los detectores inductivos, transmitir tramas de velocidad, confirmación y error hacia el nodo de configuración.

Por otra parte el nodo de configuración esta encargado de ejecutar la interacción con el usuario a fin de que esta pueda conocer el estado del motor a tratar lo que implica que desarrolle las siguientes funciones, visualizar la velocidad de cada motor emitida por el respectivo nodo de medición, enviar tramas de habilitación, confirmación y error hacia los nodos de medición, transmitir la información de activación y desactivación por el PLC hacia el nodo de medición a través del estándar de comunicaciones RS-485 y proporcionar el respectivo menú de reposición de sensores para cada nodo de medición.

### 1.3.3. Diseño e implementación de una red para control y monitoreo remoto de una planta de nivel, desarrollado por Carlos Eduardo Velásquez año 2003.

Este proyecto propone el diseño e implementación de un sistema de control de nivel en red, usando una red de microcontroladores con protocolo CAN:  
En el diseño se construyen 3 módulos que conforman el sistema:

Modulo de medición de nivel que se encarga de medir el nivel en un tanque y a su vez envía esta información por la red CAN. En el segundo modulo se hace el control de una válvula con un motor paso a paso por medio de un sistema de control difuso implementado en ensamblador para el microcontrolador. Este se encarga de manipular un conjunto de requisitos por medio de la capa de enlace del protocolo CAN:

Por ultimo encontramos el diseño y construcción de un tarjeta tipo ISA para PC que posee el software desarrollado en visual estudio 6 para hacer el monitoreo y configuración de algunos parámetros de la planta.

También existen soluciones presentadas en productos, que facilitan enormemente el manejo de diferentes tipos de señales y que representan otra posible alternativa para la solución del problema planteado.

### 1.3.4. Datalogger

Figura 7. Datalogger



Fuente: Termomed.net [en línea]. Valencia: Termomed, 2007. [Consultado 13 de noviembre. de 2007]. Disponible en Internet: <http://www.termomed.net/registrador-datos-para-estacion-p-446.html>

En la figura 7 se muestra un prototipo de registrador de datos a continuación sus principales características:

- Económico, pequeño y robusto registrador, sin tinta ni papel, sencillo de utilizar, que soporta una gran variedad de sensores para la recolección y manipuleo de variables físicas varias, tales como:

- tensión AC,
- tensión DC,
- corriente AC,
- corriente DC
- temperatura,
- presión,
- humedad,
- velocidad,
- pulsos,
- eventos

Permitiendo luego determinar el comportamiento de la variable a lo largo del tiempo, a intervalos fijos, su memorizado y posterior transferencia a una computadora portátil en el lugar de la medición o transportar el registrador para ser analizada, archivada en la PC de la oficina comercial.

1.3.5. 3710 ACM. Es un equipo de instrumentación basado en un microprocesador de 16 bit, cuya aplicación son los sistemas trifásicos industriales, comerciales, y distribuciones de energía, este monitor/medidor de potencia ofrece una destacada precisión y confiabilidad además de una excepcional resistencia. Una extensa gama para mediciones, captura de forma de onda para el análisis de armónicos, relés controlados por ajustes y además con capacidad de comunicación, hacen del 3710 ACM no tan solo una respuesta poderosa sino que también económica ante las necesidades de la administración frente a los dispositivos tradicionales de medición análogos (electro mecanismos), puesto que es capaz de reemplazar 12 transductores y mediadores individuales.

#### 1.4. OBJETIVO GENERAL

Implementar la Ingeniería básica, conceptual y de detalle para el diseño de un sistema de monitoreo de equipos de apoyo de aire acondicionado en la central telefónica de Colón.

#### 1.5. OBJETIVOS ESPECÍFICOS

- Desarrollar la Ingeniería conceptual.

- Desarrollar la Ingeniería básica: primer prototipo.
- Desarrollar Ingeniería de detalle: prototipo final.
- Realizar y evaluar dispositivo final.
- Elaborar un artículo en formato IFAC sobre el tema.
- Elaborar manuales de usuario (operación y mantenimiento).

## 1.6. JUSTIFICACIÓN

El control y monitoreo de los equipos de apoyo implementados en la central de EMCALI, le permitirán a la empresa poder tener control absoluto sobre ellos, lo cual implicará una gran ventaja en el presupuesto otorgado para el mantenimiento correctivo y preventivo de los equipos. Esto beneficiará todo el personal de mantenimiento de los equipos de apoyo, permitiendo la solución rápida en caso de fallas o averías en dichos sistemas, esto representa para la empresa mayor eficiencia en la prestación del servicio.

Con base en el desarrollo de esta estructura, se pretende lograr unificar toda la información presente en todas las centrales bajo una misma red, que permita reconocer e identificar las fallas presentes en cada central.

## 1.7. METODOLOGÍA

Para el desarrollo del proyecto utilizaremos el método secuencial. El cual esta dividido en las siguientes fases con sus respectivas actividades:

### Fase conceptual

- Estudio de mercado.
- Diagrama de flujo y/o bloques.
- Estudio económico, recursos necesarios y fuentes de financiación.
- Grado de automatización.
- Análisis de tecnologías disponibles.
- Dimensionamiento del proyecto.
- Estudio de la mano de obra disponible y requerida.
- Estudio técnico y valorización del proyecto
- Evaluación técnico económico de la factibilidad del proyecto.

### Ingeniería básica: primer prototipo

- Adquisición de Now How
- Diagramas
- Selección de normas a seguir

- Especificación de instrumentos y equipo de control
- Especificaciones de montaje
- Estrategia de control
- Diagrama localización Instrumentos
- Listado de instrumentos
- Cotización de instrumentos, equipo de control y software
- Cotización de materiales
- Cotización de montaje, pruebas y soporte en puesta en marcha.

Ingeniería de detalle: prototipo final

- Diagramas lógicos
- Detalle cuarto de monitoreo
- Configuración sistema de monitoreo
- Listado de materiales
- Especificaciones técnicas para montaje

Realizar y evaluar dispositivo final

Elaborar artículo en formato IFAC

Elaborar manual de usuario

## 1.8. CRONOGRAMA

Tabla 1. Cronograma de actividades

4	Objetivo	Duracion/Dias	Responsable	Procedencia	Recursos
5					
6	<b>Desarrollar ingeniería conceptual</b>				
7	Estudio de mercado.	1 semana	pasantes		internet, bibliografías, proyectos anteriores
8	Diagrama de flujo y/o bloques.	3 días	pasantes		normas de diseño, planos
9	Estudio económico, recursos necesarios y fuentes de financiación.	1 semana	pasantes	concluidos ítems 8,7	internet, bibliografías
10	Grado de automatización.	2 días	pasantes		estudio de la fases de automatización
11	Análisis de tecnologías disponibles.	1 semana	pasantes	concluidos ítems 8,9	catalogos, proveedores.
12	Dimensionamiento del proyecto.	2 días	pasantes		
13	Estudio de la mano de obra disponible y requerida.	2 días	pasantes	concluido ítems 10	estudio de talento humano
14	Estudio técnico y valorización del proyecto	2 semanas	pasantes	todos los ítems anteriores deben estar resueltos	análisis económico y financiero
15	Evaluación técnico económico de la factibilidad del proyecto.	3 semanas	Emcali	todos los ítems anteriores deben estar resueltos	evaluación financiera de un proyecto.
16					
17					
18	<b>desarrollar ingeniería básica: primer prototipo</b>				
19	Adquisición de Now How	1 semana	pasantes	1 primer objetivo concluido	conocimientos adquiridos durante la carrera.
20	Diagramas	1 semana	pasantes		normas de diseño
21	Selección de normas a seguir	1 semana	pasantes	desarrollado el ítem 19	conocimientos acerca de las normas establecidas
22	Especificación de instrumentos y equipo de control	1 semana	pasantes	desarrollado ítem 19 al 21	catalogos, datos técnicos, fichas bibliográficas
23	Especificaciones de montaje	2 semanas	pasantes		normas de seguridad, protección.
24	Estrategia de control	1 semana	pasantes	desarrollados ítems 19 al 23	conocimientos en control y automatización.
25	Diagrama localización Instrumentos	1 semana	pasantes	desarrollados ítems 19 al 23	normas de seguridad, protección.
26	Listado de instrumentos	1 semana	pasantes	desarrollados ítems 22,23,24	conocimiento de los materiales a implementar.
27	Cotización de instrumentos, equipo de control y software	2 semanas	pasantes	desarrollados ítems 26	comercialización con proveedores.
28	Cotización de materiales	2 semanas	pasantes	desarrollados ítems 25,26,27	comercialización con proveedores.
29	Cotización de montaje, pruebas y soporte en puesta en marcha	3 semanas	pasantes	todos los ítems anteriores deben estar resueltos	
30					
31	<b>desarrollar ingeniería de detalle: prototipo final</b>				
32	Diagramas lógicos	1 semana	pasantes	segundo objetivo concluido	conocimientos de la información a obtener
33	Detalle cuarto de monitoreo	1 semana	pasantes		mapa del lugar
34	Configuración sistema de monitoreo	1 semana	pasantes	desarrollados los anteriores	conocimientos sobre el método a implementar
35	Listado de materiales	1 semana	pasantes	desarrollados los anteriores	definido el diseño a implementar
36	Especificaciones técnicas para montaje	1 semana	pasantes	desarrollados los anteriores	definido los implementos a utilizar
37					
38	<b>Realizar y evaluar dispositivo final</b>	2 semana	pasantes	desarrollados los puntos anteriores	sistema listo para operación e implementación
39					
40					
41	<b>Elaborar artículo en formato IFAC</b>	1 semanas	pasantes	desarrollados los puntos anteriores	información de todo el proceso que conlleva a el producto
42					
43					
44	<b>Elaborar manuales de usuario</b>	1 semanas	pasantes	desarrollados los puntos anteriores	información acerca de la operabilidad del sistema

## 1.9. PRESUPUESTO

Los gastos estimados para la elaboración de este proyecto son discriminados de la siguiente manera: ver tabla 2

Tabla 2. Presupuesto

Ítem	Costo
1.Computadoras	\$3.000.000
2. Costos de Ingeniería (sensores, integrados, material electrónico, etc.)	\$2.000.000
3.Transporte	\$312.000
4. impresión y papelería	\$200.000
5. Costos de asesor UAO	\$520.000
6.Costos de Ingeniería(\$20.000/ hora de trabajo)	\$2.400.000
TOTAL	\$8.432.000

## 1.10. FINANCIACIÓN

La empresa EMCALI EICE ESP. Cubrirá todos los gastos a excepción del ítem 1 y 2 del presupuesto.

Se puede concluir que la prioridad en esta etapa preliminar de diseño es que pueda ser monitoreada desde otros puntos, para esto se prestó atención a diferentes posibilidades tecnologías, equipos, que serán analizados con mas detalle en el siguiente capítulo, también se mostrara esquemas como preámbulo a plantear para la ejecución como un producto.



## 2. INGENIERÍA BÁSICA: PRIMER PROTOTIPO

### 2.1. ADQUISICIÓN DE NOW HOW.

A continuación hablaremos acerca de cada una de las soluciones a implementar en nuestro proyecto, como será utilizada y porque se a decidido por ella.

2.1.1. Los microcontroladores. Cuando se planteó la comparación entre tecnologías, se expuso las ventajas y desventajas que tiene cada una con respecto a la otra, si embargo se ultimó que el microcontrolador fue el dispositivo que permitió implementar el diseño de forma más sencilla en muchos aspectos y sobre todo menos costoso con respecto a los otros tipos descritos.

Ahora consideraremos que plataforma de desarrollo servirá para plantear el sistema ver tabla 3, esto se hará distinguiendo las diferentes familias de dispositivos conocidas en el mercado.

Tabla 3. Comparación entre familias de microcontroladores

FAMILIAS DE MICROCONTROLADORES			
TABLA COMPARATIVA POR MARCAS DE GAMMA MEDIA			
<b>CARACTERISTICAS</b>	<b>ATMEL AT80C51</b>	<b>PIC 16F877A</b>	<b>MOTOR OLA 68HC90 8GP3</b>
Frecuencia máxima	12 MHZ	DX-20MHz	8MHZ
Memoria de programa flash palabra de 14 bits	2KBYTES	8KB	32 KBYTES
Posiciones RAM de datos	128 BYTES	368	512 BYTES
Posiciones EEPROM de datos	4 KBYTES	256	-
Puertos E/S	4	A,B,C,D,E	A,B,C,D
Número de pines	32	40	40
Interrupciones	5	14	16
Timers	2	3	2
Módulos CCP	-	2	-

Comunicaciones Serie	UART	MSSP, USART	MSSP, USART
Comunicaciones paralelo	-	PSP	-
Líneas de entrada de CAD de 10 bits	-	8	8
Juego de instrucciones	-	35 Instrucciones	40
Longitud de la instrucción	-	14 bits	16 bits
Arquitectura	-	Harvard	-
CPU	8 BITS	Risc	16 BITS
Canales Pwm	-	2	1
Pila Hardware	-	-	-
Ejecución En 1 Ciclo Máquina	-	-	-

Se puede concluir que la opción vas viable en términos económicos, accesibilidad, y desarrollo, es la que otorgan los microcontroladores microchip, aunque estos son dispositivos mas caros en comparación con los atmel, poseen una periferia superior a estos, y el software que los programa es gratis y de fácil obtención, a diferencia del Motorola, que aunque son equipos superiores a los microchip, estos no son muy accesibles por su precio, y por que su programador es muy especial, no es gratis, y es de difícil acceso.

2.1.2. Comunicaciones. Como se mostró en el diagrama de bloques, se ha considerado implementar en el diseño, 2 plataformas de comunicaciones, las cuales serán usadas para interactuar con otros dispositivos en red, en cada plataforma va implementado un protocolo de enlace, el cual es usado para facilitar el manejo del flujo de información. Investigando los diferentes dispositivos en el mercado se pudo observar que estos manejan protocolos de enlace propios o propietarios conocidos como estándares, y para poder obtener información acerca de estos hay que pagar por los derechos de uso para acceder a la configuración, también existen estándares en los cuales se puede configurar el sistema pero para que en el mercado el sistema sea reconocido como un producto bajo un estándar dado, hay que sugerirlo a unas pruebas por las que por supuesto hay que pagar, esto se expone debido a que la incorporación de este tipo de protocolos es complicada, por esto se decide escoger un protocolo abierto, lo cual significa que es una tecnología disponible lo que implica que no hay que pagar ningún costo.

- Modbus. Es un protocolo de comunicaciones situado en el nivel 7 del Modelo OSI, basado en la arquitectura maestro/esclavo o cliente/servidor, diseñado en 1979 por Modicon para su gama de controladores lógicos programables (PLCs). Convertido en un protocolo de comunicaciones estándar de facto en la industria es el que goza de mayor disponibilidad para la conexión de dispositivos electrónicos industriales. Las razones por las cuales el uso de Modbus es superior a otros protocolos de comunicaciones son:

- es público
- su implementación es fácil y requiere poco desarrollo
- maneja bloques de datos sin suponer restricciones

Modbus permite el control de una red de dispositivos, por ejemplo un sistema de medida de temperatura y humedad, y comunicar los resultados a un ordenador. Modbus también se usa para la conexión de un ordenador de supervisión con una unidad remota (RTU) en sistemas de supervisión adquisición de datos (SCADA). Existen versiones del protocolo Modbus para puerto serie y Ethernet (Modbus/TCP).

Existen dos variantes, con diferentes representaciones numéricas de los datos y detalles del protocolo ligeramente desiguales. Modbus RTU es una representación binaria compacta de los datos. Modbus ASCII es una representación legible del protocolo pero menos eficiente. Ambas implementaciones del protocolo son serie. El formato RTU finaliza la trama con una suma de control de redundancia cíclica (CRC), mientras que el formato ASCII utiliza una suma de control de redundancia longitudinal (LRC). La versión Modbus/TCP es muy semejante al formato RTU, pero estableciendo la transmisión mediante paquetes TCP/IP.

Modbus Plus (Modbus+ o MB+), es una versión extendida del protocolo que permanece propietaria de Modicon. Dada la naturaleza de la red precisa un coprocesador dedicado para el control de la misma. Con una velocidad de 1 Mbit/s en un par trenzado sus especificaciones son muy semejantes al estándar EIA/RS-485 aunque no guarda compatibilidad con este.

Cada dispositivo de la red Modbus posee una dirección única. Cualquier dispositivo puede enviar órdenes Modbus, aunque lo habitual es permitirlo sólo a un dispositivo maestro. Cada comando Modbus contiene la dirección del dispositivo destinatario de la orden. Todos los dispositivos reciben la trama pero sólo el destinatario la ejecuta (salvo un modo especial denominado "Broadcast"). Cada uno de los mensajes incluye información redundante que asegura su integridad en la recepción. Los comandos básicos Modbus permiten controlar un dispositivo RTU para modificar el valor de alguno de sus registros o bien solicitar el contenido de dichos registros.

Existe gran cantidad de modems que aceptan el protocolo Modbus. Algunos están específicamente diseñados para funcionar con este protocolo. Existen implementaciones para conexión por cable, wireless, SMS o GPRS. La mayoría de problemas presentados hacen referencia a la latencia y a la sincronización. Variaciones:

Todas las implementaciones presentan variaciones respecto al estándar oficial. Algunas de las variaciones más habituales son:

- Tipos de Datos
- Coma Flotante IEEE
- entero 32 bits
- datos 8 bits
- tipos de datos mixtos
- campos de bits en enteros
- multiplicadores para cambio de datos a/de entero. 10, 100, 1000, 256
- Extensiones del Protocolo
- direcciones de esclavo de 16 bits
- Tamaño de datos de 32 bits (1 dirección = 32 bits de datos devueltos.)

Para el desarrollo del sistema, se optó por el protocolo de comunicaciones Modbus en modo RTU, este es un protocolo muy implementado a nivel industrial, lo cual garantiza una gran gama de sistemas que podrán ser manipulados por medio de él.

- Configuración del modbus rtu. Como se explica en el punto anterior, el protocolo Modbus posee diferentes campos los cuales son usados para el fácil y seguro manejo de información, cada uno de estos campos debe ser tenido en cuenta a la hora de configurar la comunicación de esto depende la confiabilidad de la información enviada y recibida por los dispositivos.

Tabla 4. Trama del mensaje RTU

RTU Message Frame					
Start	Addr.	Function	Data	CRC	End
t1t2t3t4	8 bits	8 bits	nx8 bits	16 bits	t1t2t3t4

En la tabla 4 se muestran los campos y la extensión de cada uno.

El mensaje RTU empieza con un intervalo de silencio de al menos 3.5 veces el tiempo de transmisión de un carácter a una velocidad de transmisión designada para la red (mostrada en la tabla como t1t2t3t4). El primer campo transmitido es la dirección del dispositivo (address). Los caracteres permitidos para transmitir para todos los campos son valores hexagesimales 0-9, A-F.

Un dispositivo en red esta monitoreando continuamente incluyendo los intervalos de silencio, y cuando el primer campo es recibido (address) después

de un intervalo de 3.5 el tiempo por carácter, el dispositivo lo decodifica para verificar de que es la dirección del dispositivo seguido del último carácter transmitido, un intervalo de silencio similar al anterior marca el fin del mensaje después del intervalo, puede empezar otro mensaje.

El mensaje completo debe ser enviado como una cadena completa si un intervalo de 1.5 veces el tiempo por carácter ocurre antes de que se complete la trama (no una trama completa), el dispositivo vota el mensaje incompleto y asume que el siguiente carácter será la dirección.

De forma similar, si un Nuevo mensaje empieza antes de 3.5 veces el tiempo por carácter, el dispositivo que recibe asumirá que este es una continuación del mensaje anterior. Lo cual genera un error, el valor del CRC no será igual por la combinación de los 2 mensajes.

Cada carácter es transmitido de tal manera que está dividido en 2 un nibble alto y un nibble bajo. Abajo se muestra como es la ventana de transmisión de cada carácter.

Tabla 5. Transmisión de caracteres

RTU Character Framing (No Parity)											
Start	0	1	2	3	4	5	6	7	Stop	Stop	
RTU Character Framing (With Parity)											
Start	0	1	2	3	4	5	6	7	Parity	Stop	

En el diseño del proyecto se usan 2 bits de stop para facilidad en la transmisión ver tabla 5.

El tiempo por carácter utilizado en el sistema, se calcula con base en que la red trabajará a una velocidad de 2400 bauds, Se escogió esta velocidad debido a que la aplicación no requiere altas velocidades de transmisión.

Un dato transmitido son 11 bits como mostramos anteriormente  
Esto significa que 11 bit/dato con base esto se tiene que:

Formula 1. Calculo del tiempo de envío de una palabra Modbus

$$Time = \frac{11 \text{ bits/char}}{2400 \text{ bits/sec}} = 4583.333 \mu\text{s/char}$$

$$T_{frame} = 3.5 * Time$$

$$T_{frame} = 16041.666 \mu\text{s/char}$$

$$T_{in} = 1.5 * Time$$

$$T_{in} = 6875 \mu\text{s/char}$$

Estos serán los tiempos a tener en cuenta para el desarrollo de la plataforma Modbus RTU tipo maestro, hay que agregar que siendo el dispositivo en desarrollo el maestro, estos tiempos no son tan críticos como los serían en el esclavo, sin embargo es mejor tenerlos en cuenta para el desarrollo de esta.

Después de presentar la configuración básica del Modbus, se procede a seleccionar los mandos que ejecutará la plataforma, mas conocidas como funciones, estas son un set de instrucciones las cuales indican como se debe configurar el mensaje para lograr un propósito en especial, en la tabla 6 se muestra las principales:

Tabla 6. Set de instrucciones más utilizadas

CODE	FUNCTION	REFERENCE
01 (01H)	Read Coil (Output) Status	0xxxx
03 (03H)	Read Holding Registers	4xxxx
04 (04H)	Read Input Registers	3xxxx
05 (05H)	Force Single Coil (Output)	0xxxx
06 (06H)	Preset Single Register	4xxxx
08 (08H)	Reset Slave	Hidden
15 (0FH)	Force Multiple Coils (Outputs)	0xxxx
16 (10H)	Preset Multiple Registers	4xxxx
17 (11H)	Report Slave ID	Hidden

En la implementación se decidió utilizar las primeras 5 funciones y la función 15 expuestas en la tabla 6, se considera que las demás son de un uso muy específico que no es requerido para la solución, y que quizás sean implementadas en futuras versiones del producto, a continuación se explican cada función usada y su uso en el proyecto.

- **Read Coil Status:** esta función permite leer un registro específico del sistema y conocer el estado de sus entradas discretas o bobinas. La configuración de esta es fija solo hay que añadirle la dirección del esclavo, la función (01) y el registro a leer ver tabla 7.

Tabla 7. Lectura del estado espiral

**Read Coil Status Example Query**

Field Name	Example Value (Hex)
Slave Address	247 (F7)
Function Code	1 (01)
Starting Address High Order	0 (00)
Starting Address Low Order	0 (00)
Number Of Points High Order	0 (00)
Number Of Points Low Order	4 (04)
CRC Error Check (LRC in ASCII Mode)	--

- **Read Holding Register:** esta función se utiliza para leer varios registros a la vez con dirección de forma 4x, se configura con la dirección del esclavo, la función (03), el registro inicial y el registro final ver tabla 8.

Tabla 8. Lectura de registros 4x

**Read Holding Register Example Query**

Field Name	Example Value (Hex)
Slave Address	247 (F7)
Function Code	3 (03)
Starting Address High Order	0 (00)
Starting Address Low Order	5 (05)
Number Of Points High Order	0 (00)
Number Of Points Low Order	3 (03)
CRC Error Check (LRC in ASCII Mode)	--

- **Read Input Registers:** de forma similar que la anterior, esta función se utiliza para leer varios registros a las ves con dirección de forma 3x, se configura con la dirección del esclavo, la función (04), el registro inicial y el registro final ver tabla 9.

Tabla 9. Lectura de registros 3x

Read Input Registers Example Query

Field Name	Example Value (Hex)
Slave Address	247 (F7)
Function Code	4 (04)
Starting Address High Order	0 (00)
Starting Address Low Order	2 (02)
Number Of Points High Order	0 (00)
Number Of Points Low Order	2 (02)
CRC Error Check (LRC in ASCII Mode)	--

- **Force Single Coil:** Se usa para cambiar el estado de una sola salida, on/off esta función puede ser usada de forma broadcast (todos los elementos conocidos y desconocidos de la red la pueden ejecutar), solo requiere la función (05) la dirección de la bobina y el estado al que se quiere cambiar ver tabla 10.

Tabla 10. Lectura de estado ON/OFF

Force Single Coil Example Query And Response

Field Name	Example Value (Hex)
Slave Address	247 (F7)
Function Code	5 (05)
Coil Address High Order	0 (00)
Coil Address Low Order	3 (03)
Force Data High Order	255 (FF)
Force Data Low Order	0 (00)
CRC Error Check (LRC in ASCII Mode)	--

- **Preset Single Register:** esta función se usa para dar un valor específico a un registro específico, esta función puede ser usada de forma broadcast (todos los elementos conocidos y desconocidos de la red la pueden ejecutar), requiere la dirección del esclavo, la función, los registros a configurar y la información a enviar ver tabla 11.

Tabla 11. Valor específico a registros

Preset Holding Register Example Query (And Response)

Field Name	Example Value (Hex)
Slave Address	247 (F7)
Function Code	6 (06)
Register Address High Order	0 (00)
Register Address Low Order	1 (01)
Preset Data High Order	0 (00)
Preset Data Low Order	2 (02)
CRC Error Check (LRC in ASCII Mode)	--

- **Force Multiple Coils:** funciona exactamente que la función 05(Force Single Coil) solo que esta me permite modificar el estado de varios registros, también puede ser usada de forma broadcast, esta recibe la dirección del



esclavo, la función, y los registros a modificar, junto con el estado a cambiar ver tabla 12.

Tabla 12. Modificación de varios registros

Force Multiple Coils Example Query

Field Name	Example Value (Hex)
Slave Address	247 (F7)
Function Code	15 (0F)
Coil Address High Order	0 (00)
Coil Address Low Order	0 (00)
Number Of Coils High Order	0 (00)
Number Of Coils Low Order	4 (04)
Byte Count	01
Force Data High (First Byte)	5 (05)
Error Check (LRC or CRC)	--

- Protocolo ASCII. Este protocolo como dice su nombre utiliza los caracteres ASCII para hacer el reconocimiento de la información, esto permite una fácil identificación de tramas, y transmisión sencilla, en nuestro caso hemos acordado que el protocolo que se hará será configurable, es decir que el usuario decide como son las tramas de transmisión y recepción , lo que hará el sistema electrónico solo será recoger la información y enviarla, no hará análisis de esta, de eso se encargara el CPU o dispositivo de monitoreo.

- Protocolos de red. En de los puntos a solucionar que se planteo en el problema, es la posibilidad de poder acceder a la información que entrega la tarjeta desde un punto distinto al sistema central (cpu central), para esto coincidimos que la mejor forma de solucionar esto era permitiendo al sistema que pueda ser acezado por la red de la empresa, para esto el sistema debe trabajar con protocolos de red ampliamente conocidos, y que actualmente se encuentra en la mayoría de empresas del país, estos son el ethernet y hablando en forma mas general el Internet, adecuando nuestro sistema a estas 2 herramientas ampliamente distribuidas, podremos cumplir con ese requerimiento de una forma muy eficaz, si se decide utilizar o crear una red, esto puede traer muchos problemas, construir una red independiente solo para el acceso de estos dispositivos no es un opción, y usar una red ya implementada generaría problemas debido a que seria congestionar aun mas la red, que ya de por si esta congestionada, debido a estas razones es que se decide acoplar al diseño un sistema que pueda entregar la información a la red por medio de ethernet, a continuación se explica en detalle esto.

- Ethernet. Es el nombre de una tecnología de redes de computadoras de área local (LANs) basada en tramas de datos. El nombre viene del concepto físico de *ether*. Ethernet define las características de cableado y señalización de nivel físico y los formatos de trama del nivel de enlace de datos del modelo OSI. Ethernet se refiere a las redes de área local y dispositivos bajo el estándar

IEEE 802.3 que define el protocolo CSMA/CD, aunque actualmente se llama Ethernet a todas las redes cableadas que usen el formato de trama descrito más abajo, aunque no tenga CSMA/CD como método de acceso al medio. Ethernet es la capa física más popular de la tecnología LAN usada actualmente, que fue desarrollada principalmente por la empresa XEROX. Otros tipos de LAN incluyen Token Ring 802.5, Fast Ethernet, FDDI, ATM y LocalTalk. Ethernet es popular porque permite un buen equilibrio entre velocidad, costo y facilidad de instalación. Estos puntos fuertes, combinados con la amplia aceptación en el mercado y la habilidad de soportar virtualmente todos los protocolos de red populares, hacen a Ethernet la tecnología ideal para la red de la mayoría de usuarios de la informática actual. El estándar original IEEE 802.3 estuvo basado en la especificación Ethernet 1.0 y era muy similar. El documento preliminar fue aprobado en 1983 y fue publicado oficialmente en 1985 (ANSI/IEEE Std. 802.3-1985). Desde entonces un gran número de suplementos han sido publicados para tomar ventaja de los avances tecnológicos y poder utilizar distintos medios de transmisión, así como velocidades de transferencia más altas y controles de acceso a la red adicionales. Los elementos en una red Ethernet son los nodos de red y el medio de interconexión. Dichos nodos de red se pueden clasificar en dos grandes grupos: Equipo Terminal de Datos (DTE) y Equipo de Comunicación de Datos (DCE). Los DTE son los dispositivos que generan o son el destino de los datos, tales como las computadoras personales, las estaciones de trabajo, los servidores de archivos, los servidores de impresión, todos son parte del grupo de estaciones finales. Mientras que los DCE son los dispositivos de red intermediarios que reciben y retransmiten las tramas dentro de la red, y pueden ser ruteadores, conmutadores (switch), concentradores (hub), repetidores, o interfaces de comunicación, como un módem o una tarjeta de interfase por ejemplo.

- NIC, o Tarjeta de Interfaz de Red - permite el acceso de una computadora a una red local. Cada adaptador posee una dirección MAC que la identifica en la red y es única. Una computadora conectada a una red se denomina nodo.
- Repetidor o repeater - aumenta el alcance de una conexión física, recibiendo las señales y retransmitiéndolas, para evitar su degradación a lo largo del medio de transmisión, lográndose un alcance mayor. Usualmente se usa para unir dos áreas locales de igual tecnología y sólo tiene dos puertos. Opera en la capa física del modelo OSI.
- Concentrador o hub - funciona como un repetidor, pero permite la interconexión de múltiples nodos, su funcionamiento es relativamente simple, recibe una trama de ethernet y la repite por todos sus puertos, sin llevar a cabo ningún proceso sobre las mismas. Opera en la capa física del modelo OSI.

- Puente o bridge - interconectan segmentos de red, haciendo el cambio de *frames* (tramas) entre las redes de acuerdo con una tabla de direcciones que dice en qué segmento está ubicada una dirección MAC.  
Conexiones en un switch Ethernet

Conmutador o Switch - funciona como el *bridge*, pero permite la interconexión de múltiples segmentos de red, funciona en velocidades más rápidas y es más sofisticado. Los switches pueden tener otras funcionalidades, como *redes virtuales* y permiten su configuración a través de la propia red. Su funcionamiento básico es en las capas física y de enlace de datos del modelo OSI, por lo cual son capaces de procesar información de las tramas; siendo su funcionalidad más importante las tablas de dirección. Por ejemplo, una computadora conectada al puerto 1 del conmutador envía una trama a otra computadora conectada al puerto 2, el *switch* recibe la trama y la transmite a todos sus puertos, excepto aquel por donde la recibió, la computadora 2 recibirá el mensaje y eventualmente lo responderá, generando tráfico en el sentido contrario, por lo cual ahora el *switch* conocerá las direcciones MAC de las computadoras en el puerto 1 y 2, y cuando reciba otra trama con dirección de destino a alguna de ellas, sólo transmitirá la trama a dicho puerto, lo cual disminuye el tráfico de la red y contribuye al buen funcionamiento de la misma.

Para la solución del problema no se implementara protocolo ETHERNET TCP/IP, lo que se requiere es un equipo que haga la conversión de RS232 A ETHERNET TCP/IP para ello existen diferentes equipos que pueden cumplir con ese propósito.

2.1.3. Comunicaciones nivel físico. Como se definió anteriormente, se utilizara el protocolo Modbus RTU maestro a una velocidad de transmisión de 2400 bauds, conociendo esto se procede a la selección de la plataforma física donde se intercambiara la información y por la cual el Modbus se transmitirá.

El protocolo Modbus se puede implementar en diferentes tipos de niveles físicos, se utilizara para la solución los 2 puertos, tanto el Modbus como el ASCII, una red RS-232 por la cual se hará la transmisión de la información a la cual se accederá por medio de interfases RS-232/RS-485, a continuación se explica en detalle un poco del RS-485, del RS-232 y como se acoplan a dicho proyecto.

- El EIA/tia-232-e estándar. Es una interfaz (también conocido como EIA RS-232C) que designa una norma para el intercambio serie de datos binarios entre un DTE (Equipo Terminal de datos) y un DCE (Equipo de terminación del circuito de datos).

El RS-232 consiste en un conector tipo DB-25 de 25 pines, aunque es normal encontrar la versión de 9 pines DB-9, más barato e incluso más extendido para cierto tipo de periféricos (como el ratón serie del PC).

- Características eléctricas. La sección de características eléctricas del estándar RS-232 Incluye especificaciones de niveles de voltaje, tasa de cambio en los niveles de la señal, e impedancia de línea. El estándar RS-232 original fue definido en 1962. Esto fue antes de los días de la lógica TTL, no debería ser sorpresa que el estándar no use niveles de 5v y de tierra. Al contrario, un nivel alto de la salida del driver es definido de +5 to +15 v y un bajo nivel para la salida del driver es definida entre -5 y -15 v. los niveles lógicos del receptor fueron definidos para proveer un margen de ruido de 2 v. así como, un nivel alto para el receptor fue definido como +3 hasta +15 v y un nivel bajo es entre -3 hasta -15 v.

- El estándar EIA RS-485 data transmisión. Cuando se requieren mayores distancias y velocidades de transmisión, entonces deben de emplearse las normas RS422 y RS485. Además, estas normas permiten también la transmisión multipunto, es decir una computadora central conectada con varias UTR. Dado que la computadora central típicamente tiene como salida la interfaz RS232, se hace necesaria la conexión de un módulo convertidor RS232 a RS422/485, objeto del presente proyecto, para implementar una red ver tabla 13.

La transmisión diferencial permite velocidades de hasta 10 Mbps, sobre distancias de hasta 1.3 Km. Se usan dos señales para transmitir y dos para recibir, además de la tierra, la cual es normalmente conectada al blindaje del cable. En cada par, viajan la señal de transmisión y su complemento. En el receptor, la señal original se obtiene restando una de la otra. Esta técnica reduce grandemente el ruido generado en la línea, éste se induce por igual en ambas líneas del par y es al final cancelado. Este tipo de transmisión debe de hacerse siempre sobre cable del tipo "par trenzado" (twisted pairs).

Tabla 13. Especificaciones RS-485

Especificaciones		RS-485
Modo de operación		DIFERENTIAL
Máximo numero de drivers y receptores en línea(un driver a la ves para redes 485)		32DRIVER 32 RECVR
Tamaño máximo del cable		4000 FT.
Máxima tasa de datos		10Mb/s-100Kb/s
Máximo voltaje de salida del driver		-7V to +12V
Nivel de salida de la señal de l driver(min. carga)	Con carga	+/-1.5V
Nivel de salida de la señal del driver máx. carga)	Sin carga	+/-6V
Impedancia de carga del driver (Ohms)		54
Máx. corriente del driver en alta impedancia	prendida	+/-100uA
Máx. corriente del driver en alta impedancia	apagada	+/-100uA
Slew Rate (Max.)		N/A
Rango de voltaje de entrada del receptor		-7V to +12V
Sensibilidad de entrada del receptor		+/-200mV
Resistencia de entrada del receptor (Ohms)		>=12k

Fuente: RS485, RS232, RS425 [en línea]. Ohio: rs485.com., 2005. [Consultado 14 de noviembre. de 2007]. Disponible en Internet: <http://www.rs485.com/rs485spec.html>

2.1.4. Alimentación del sistema. La pretensión con este desarrollo es mas que solucionar un problema, es entregar un producto , el cual pueda ser utilizado en situaciones similares a las que generaron el diseño original, debido a esto es

necesario convertir el producto en un sistema independiente el cual opere sin la necesidad de adecuaciones especiales, para poder cumplir este objetivo, al sistema se le desarrolló su propio sistema de alimentación, el cual obtendrá su energía de la línea convencional, la rectificara y la dejara en condiciones para que pueda ser usada por la electrónica implementada en el desarrollo.

Existen diferentes métodos, los cuales proporcionan soluciones eficientes:

- Fuente controlada por PWM.
- Fuente controlada por ángulo de fase.
- Fuente rectificada por diodos.

Se considero que las 2 primeras soluciones son muy ostentosas y aunque son muy buenas, se considera que seria una solución muy poco beneficiosa, pues solo se desea entregar un nivel fijo de voltaje, no se requiere control en el sistema, por eso la tercera solución el puente rectificador, implementado con un regulador de voltaje es la solución mas factible en nuestro caso.

2.1.5. Adaptación electrónica de señales analógicas, digitales y manejo de tierras. Para esta parte se necesita considerar que el dispositivo será conectado en diferentes redes eléctricas, las cuales, pueda o no estén en buen estado, por esta razón se piensa que el dispositivo tenga la posibilidad de poder protegerse contra cualquier tipo de daño en la red eléctrica, ya sea en la parte de alimentación o en la parte de las entradas del sistema.

Para esto ya existen normas que establecen diferentes tipos de márgenes que ayudará a proteger el equipo y por supuesto el del cliente, esta parte se tratará de mejor manera cuando se definan las normas que se tendrán en cuenta para el desarrollo del proyecto.

2.1.6. Copilador de C para microcontroladores PIC. Se utilizo el compilador de C que trae el MPLAB, un software que permite programar microcontroladores del la Microchip, es gratis y de muy fácil implementación, este también permite la simulación de los proyecto, y si se posee una tarjeta de programación, también permite la capacidad de programar el microcontrolador que se desee.

2.1.7. Cableado estructurado. El solo hecho de poder compartir información es el pilar básico para la buena organización de una empresa, se refiere a comunicar información a través de una red, no todos deben tener la misma estructura organizacional, cada compañía tendrá una serie de necesidades distintas.

Por lo tanto la elección de los componentes de red (ordenadores, cableado, tarjetas de red, software, etc.) será en función del resultado que desea tener la empresa o institución.

Los elementos que configuran la red son:

- Servidor: Es el ordenador principal de la red que soporta el sistema gestor de la misma, proporcionando todos los servicios a los terminales conectados a la red.
- Estaciones de trabajo: Componen los nodos terminales de la red, pudiendo ser tan solo terminales o bien PC's, con sus propios dispositivos físicos.
- Tarjetas de red: Es el elemento interfaz que permite la conexión de una PC u ordenadora a la red, constituyendo el pilar sobre el que sostiene todo la red.

✓ Tipos de medios de transmisión

Ver anexo. 1

## 2.2. DIAGRAMAS

A continuación se mostrará los diagramas de los diferentes sistemas que se esperan implementar en el diseño.

- Diagrama de bloques del sistema en general. Se escogió Modbus RTU debido a que es un protocolo abierto de muy fácil adquisición, es muy sencillo encontrar información en la red, y además es un protocolo de mucho uso en la industria ver figura 8. Esto implica que no hay que pagar, por algún chip o reglamentación que lo determine, este protocolo quizá no sea el de mayor difusión en el campo tecnológico pero es muy tenido en cuenta por las compañías a la hora de diseñar sus productos, además es fácil de implementar en sistemas micro controlados.

Figura 8. Diagrama de bloques del sistema en general

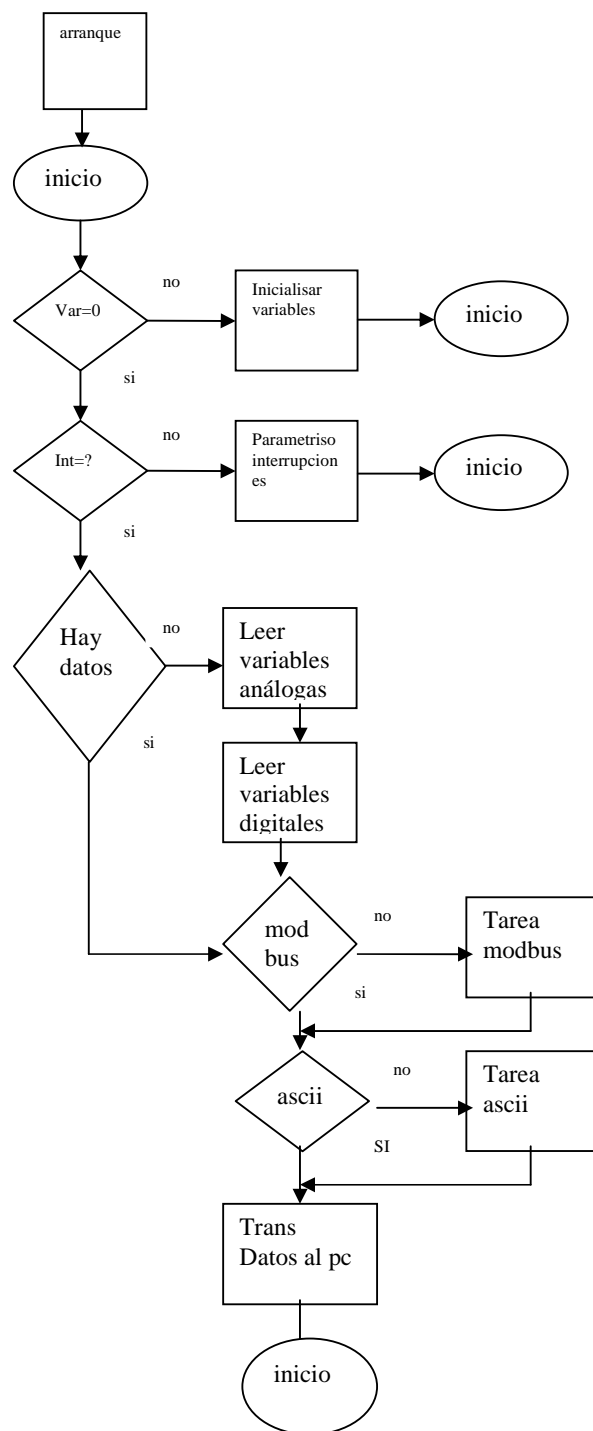
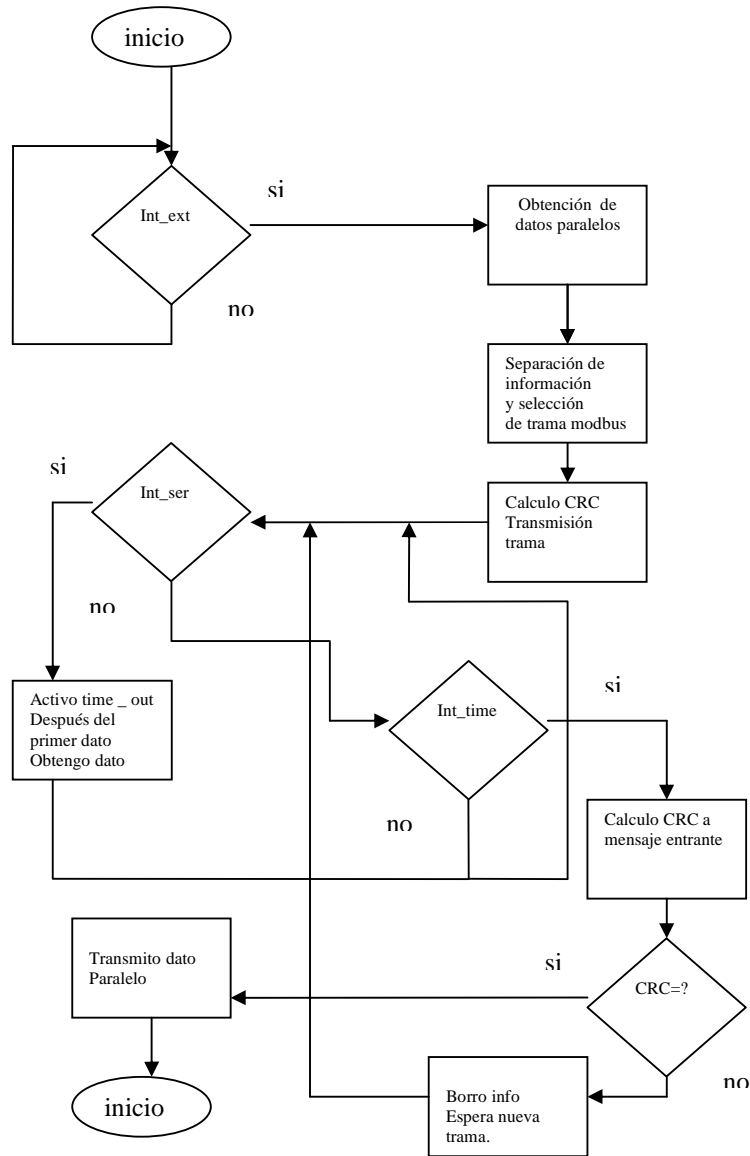




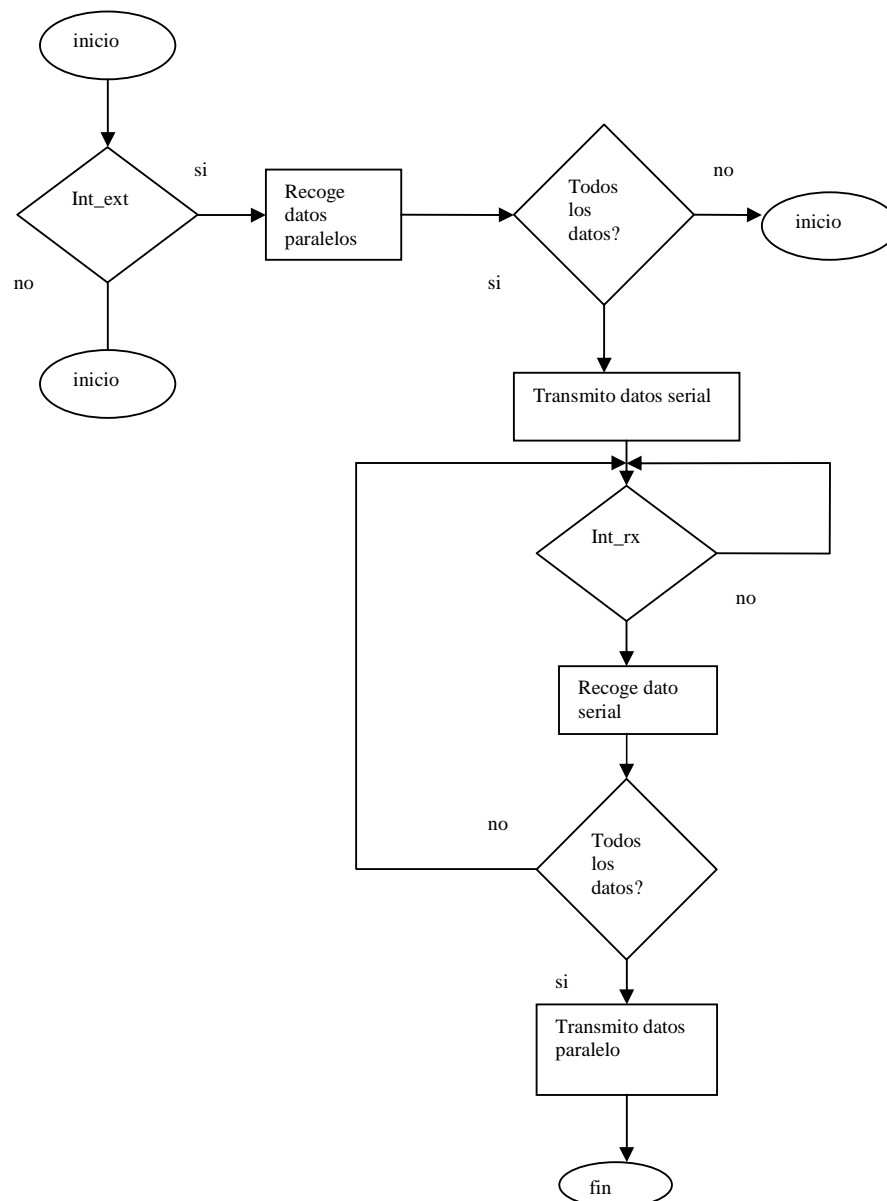
Figura 9. Diagrama de bloques del modbus rtu maestro



El protocolo Modbus posee varias versiones de las cuales se pensó en 2 para desarrollar en el proyecto , el Modbus ASCII y el Modbus RTU, se escogió este debido a la velocidad en términos de procesar la información y de transmitirla, haciendo referencia a la diferencia entre los 2 protocolos la cual consiste que el tamaño de el dato transmitido serialmente es diferente en ambos casos ver figura 9, el ASCII maneja un dato de tamaño de 7 bits, el rtu es de 8 bits, no parece una gran diferencia pero es abismal en términos de la cantidad de datos transmitidos debido a que el ASCII solo maneja caracteres de 7 bits, este solo puede manejar caracteres ASCII legibles por el sistema, al contrario del rtu que al poseer 8 bits puede manejar todos las caracteres ASCII conocidos, citando un ejemplo; si se desea enviar por ejemplo el numero 200 en ascii, se

tendría que dividir en tres partes y enviarlas por separado así 2 , 0 , 0 esto debido a que 7 bits solo me permite enviar hasta el 128, en cambio en rtu es hasta 255, lo cual significa que simplemente envié el equivalente ASCII de 200, por eso su gran diferencia.

Figura 10. Diagrama de bloques sistema serial ASCII

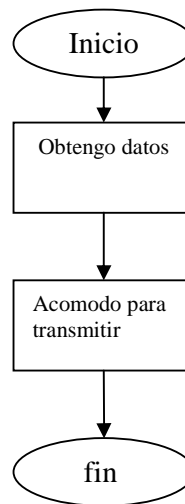


Este protocolo como dice su nombre utiliza los caracteres ASCII para hacer el reconocimiento de la información ver figura 10, esto permite una fácil identificación de tramas, y transmisión sencilla. Esto permite que el protocolo que se hará será configurable, es decir que el usuario decide como son las tramas de transmisión y recepción, lo que hará el sistema electrónico solo será

recoger la información y enviarla, no hará análisis de esta, de eso se encargara el CPU o dispositivo de monitoreo.

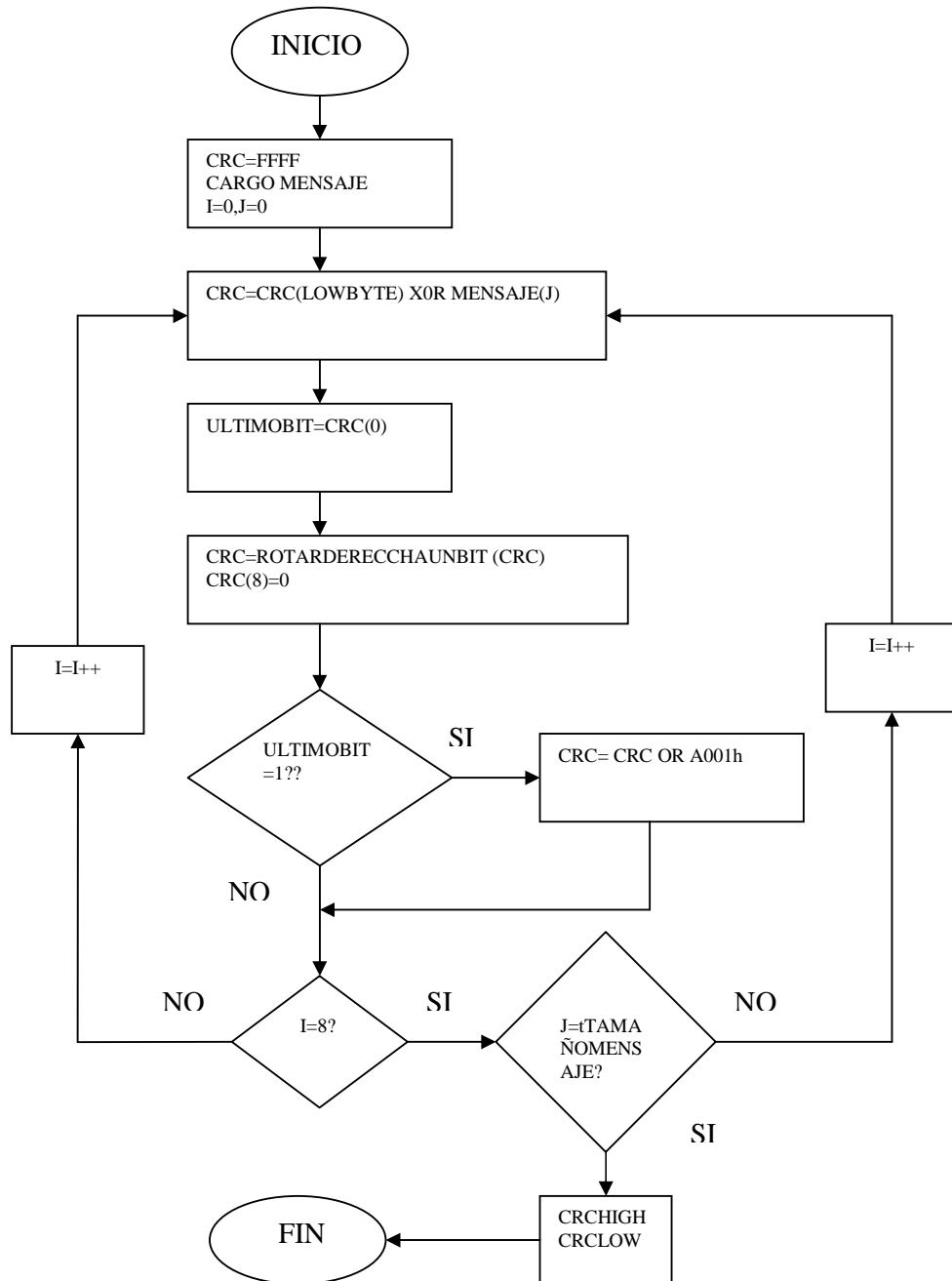
En ambos puertos el de monitoreo y el de transmisión de datos, se maneja un protocolo ASCII, solo que diferirán en los campos y en la velocidad de transmisión, que se definió de 2400 bauds en el ASCII configurable, y de 9600 en el intercambio de información con la PC, se determinó estas velocidades debido a que en el primer caso no se requieren grandes velocidades los sistemas no van a estar a más de 20 o 60 m.

Figura 11. Diagrama de bloque del sistema que obtiene datos analógicos y digitales



Como ya se especificó el sistema poseerá 6 entradas analógicas y 8 digitales, exclusivas para monitoreo ver figura 11, también como se había ya descrito, los sistemas de control de la central de EMCALI son todos iguales y trabajan según las normas estándares de señales, para lo cual las entradas deben ser creadas de acuerdo a estas normas para así garantizar un producto que se puede implementar en cualquier área de trabajo reglamentada. Se explica con más detalle en el trabajo principal este punto.

Figura 12. Diagrama del sistema de cálculo del CRC.



El CRC (Código de Redundancia Cíclica) ver figura 12 es una parte importante del Modbus la cual se encarga de verificar la veracidad de la información, esto lo hace mediante un código el cual genera byte por byte del mensaje, una especie de firma la cual es comparada, como se explico anteriormente esta viene incluida en la trama Modbus, hay diferentes métodos para crear este código, se utilizó uno recomendado por la empresa Modicon que son los creadores del protocolo.

Figura 13. Diagrama del sistema transmisor paralelo (bus de cuatro bits) recepción.

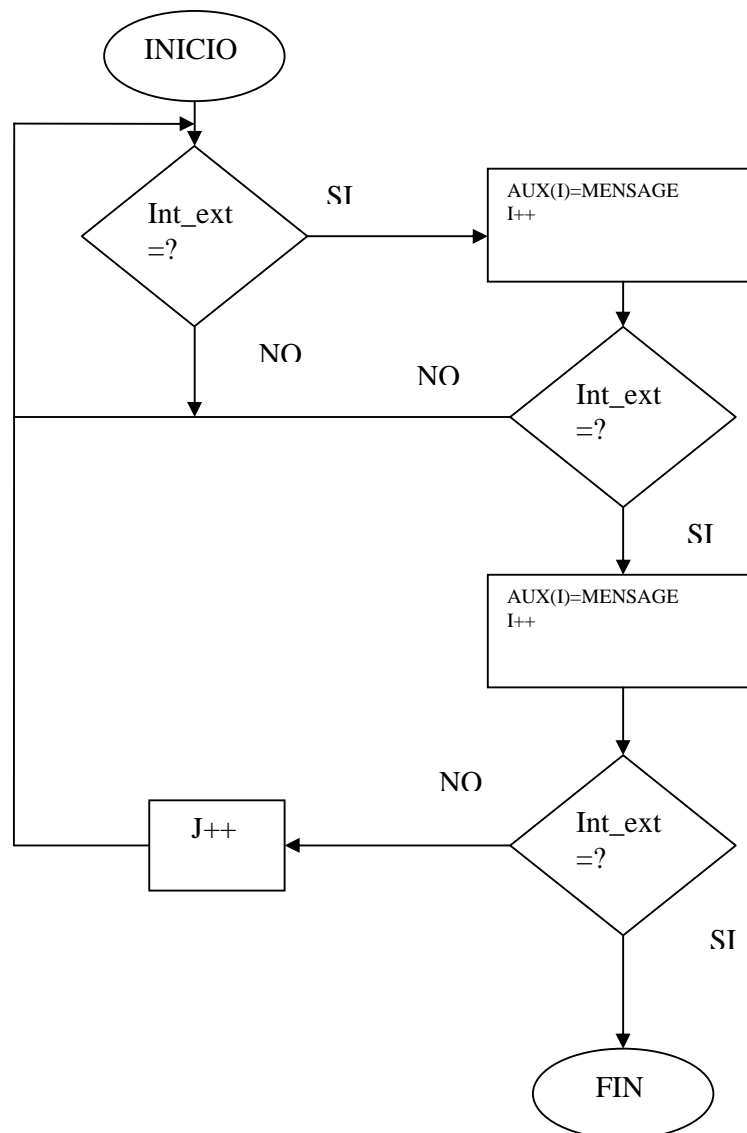


Figura 14. Diagrama del sistema transmisor paralelo (bus de cuatro bits) transmisión

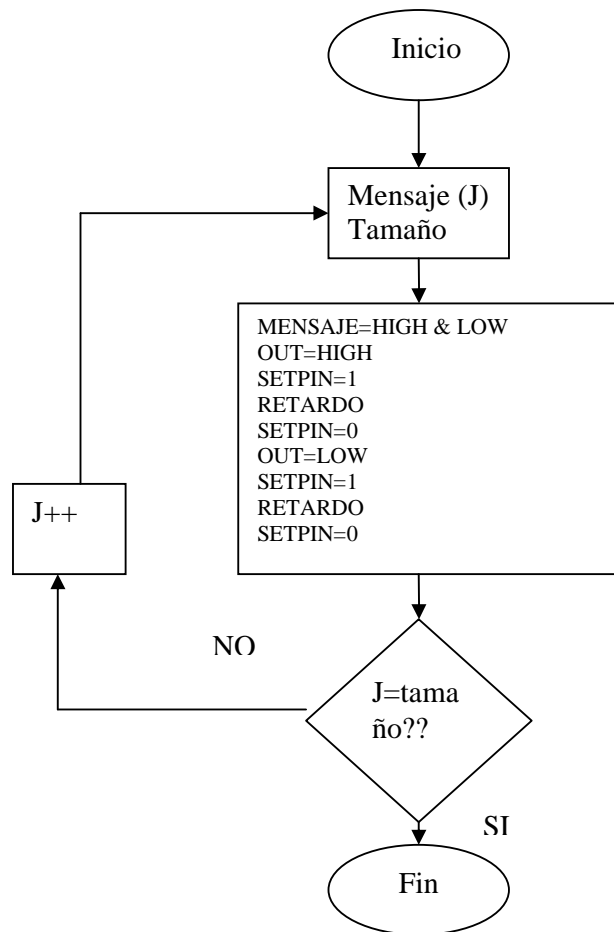
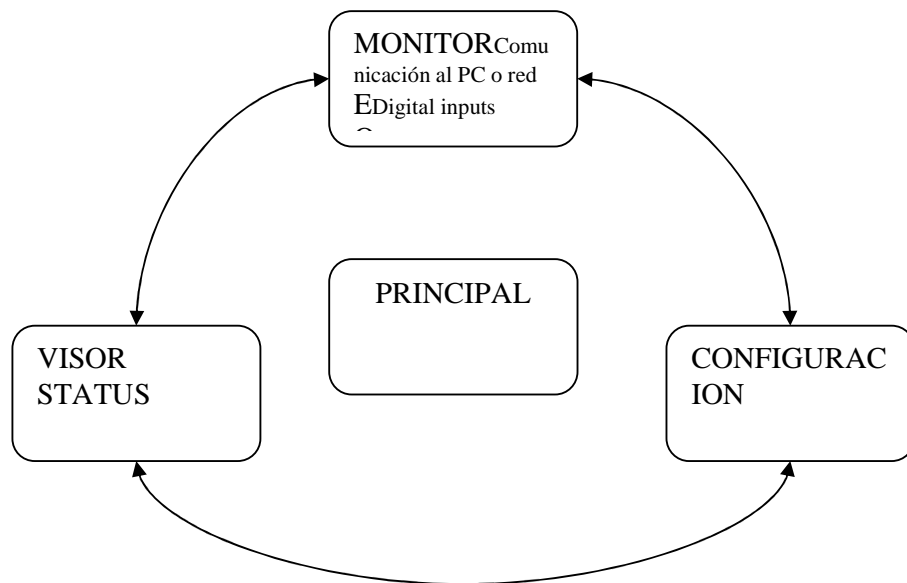


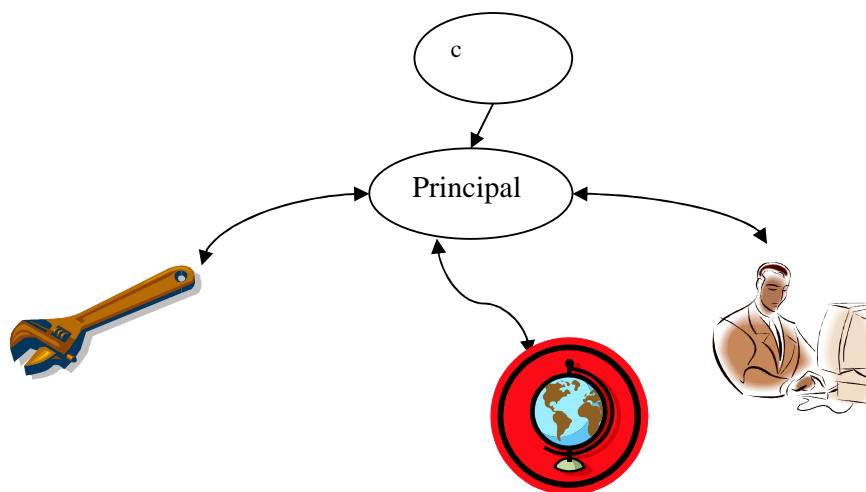
Figura 15. Diagrama del software principal



Este software es para poder tener acceso al dispositivo, con este fin como se había planteando, se procede a utilizar el compilador de código Visual Basic, el cual con sus diferentes posibilidades, permite generar programas con muchas aplicaciones.

Esta arquitectura se presenta como opción de diseño de software ver figura 15, el cual tiene diferentes tipos de alternativas que ayudan al operario en su trabajo.

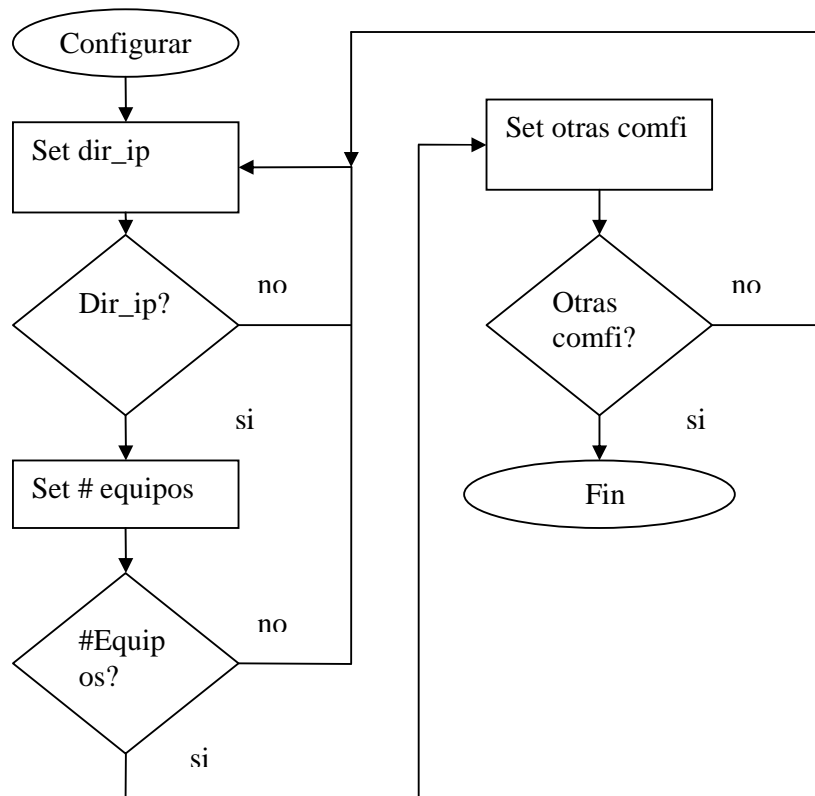
Figura 16. Arquitectura de software



A continuación se explicará cada una de las partes del software a implementar.

- Diagrama de funcionamiento de la sección de configuración. El usuario debe conocer la existencia de otras tarjetas que estén instaladas en otras centrales, para así saber como se debe configurar el software, también debe conocer por cada tarjeta cuantos equipos hay conectados, tanto en la red ASCII, como en la red modbus, Se debe aclarar que en esta etapa , no se ha definido la cantidad de equipos que se podrán acoplar a las redes ASCII o modbus, esto es debido a que no se ha definido la capacidad de la tarjeta, ya sea eléctricamente, en términos de cantidad de información manejada por el dispositivo.

Figura 17. Diagrama de funcionamiento de la sección de configuración

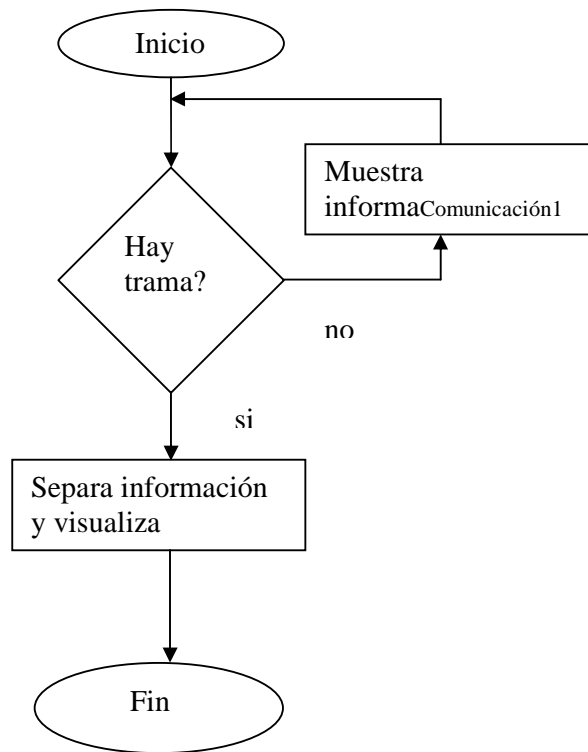


En la figura 17 se muestra una parte que dice “otras confi”, es para la posibilidad de agregar otro tipo de configuraciones mientras se hace el diseño.



- Diagrama de funcionamiento de la sección de visor status. En esta parte se le permite al usuario poder observar el comportamiento del sistema al cual se le desea ver su status., esta parte esta directamente relacionada con la sección de monitoreo.

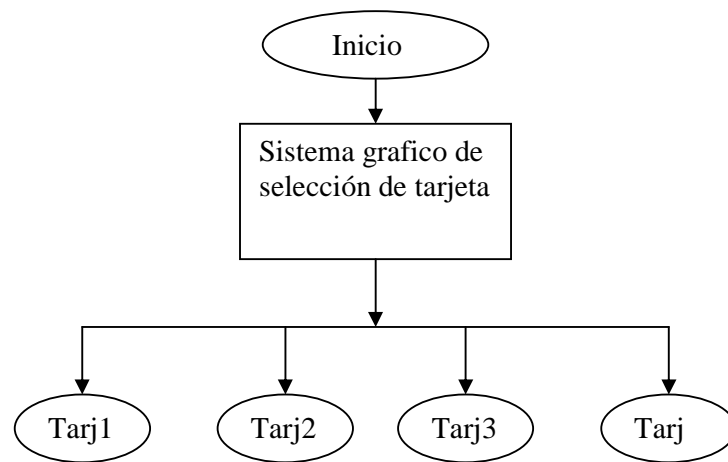
Figura 18. Diagrama de funcionamiento de la sección de visor status



Esta información que se guarda, es clasificada por equipos, fecha, hora, etc., ver figura 18

- Diagrama de funcionamiento de la sección monitor. En esta sección lo que se hace es mostrar los diferentes tarjeta que actualmente se encuentran a disposición en la red (Internet), aquí se puede escoger la tarjeta a la cual se le desea ver el status.

Figura 19. Diagrama de funcionamiento de la sección monitor



Como se muestra en la figura 19 no se especifica un numero fijo de equipos a conectar, esta parte no se especificara aun, debido a que no se sabe con exactitud, un numero mínimo de centrales, que requiera este sistema.

## 2.3. SELECCIÓN DE NORMAS A SEGUIR

2.3.1. Simbología isa y sama. Para designar y representar los instrumentos de medición y control se emplean normas muy variadas que a veces varían de industria en industria. Esta gran variedad de normas y sistemas utilizados en las organizaciones industriales indica la necesidad universal de una normalización en este campo. Varias sociedades han dirigido sus esfuerzos en este sentido, y entre ellas se encuentra como una de las importantes la Sociedad de Instrumentos de Estados Unidos, ISA (Instrument Society of America) cuyas normas tienen por objeto establecer sistemas de designación (código y símbolos) de aplicación a las industrias químicas, petroquímicas, aire acondicionado, etc.

Ver anexo 2. Selección de normas a seguir.

2.3.2. Norma ul. Underwriters Laboratories Inc. (UL) es un organismo independiente de pruebas ensayos de seguridad y certificación, que evalúa productos, materiales y sistemas trabajando por la seguridad de los consumidores y sus bienes desde 1894. UL es líder mundial en el desarrollo de normas de seguridad de productos, las cuales son frecuentemente actualizadas

y revisadas para responder a los cambios en la tecnología o responder a nuevos usos de estos productos.

2.3.3. Norma retie. El Reglamento Técnico para Instalaciones Eléctricas - RETIE, contempla las obligaciones y responsabilidades de todos los actores involucrados en los procesos de generación, transmisión, transformación, distribución y Uso final de la energía eléctrica. Desde el 30 de abril de 2005 el RETIE se encuentra vigente.

2.3.4. Norma IP (internacional protection).. Las clases de protección IP (IP = International Protection) se definen en las normas DIN VDE 0470 y DIN 40 050. Estas normas clasifican las clases de protección de los elementos que albergan equipos eléctricos cuyas tensiones no sobrepasen los 72,5 kV. Concretamente, las normas definen lo siguiente:

- Protección de personas: Protección de equipos eléctricos contra la penetración de cuerpos extraños sólidos, incluyendo el polvo. Protección de equipos eléctricos contra efectos dañinos ocasionados por agua.
- Explicación código IP. El código IP está compuesto de las letras «IP» y de dos cifras. La primera de ellas puede ser un número entre 0 y 6 y la segunda

2.3.5. Estándar IEEE 802.3. IEEE 802.3 es el nombre de un comité de estandarización del IEEE y por extensión se denominan así los estándares por él producidos.

La primera versión fue un intento de estandarizar ethernet aunque hubo un campo de la cabecera que se definió de forma diferente, posteriormente ha habido ampliaciones sucesivas al estándar que cubrieron las ampliaciones de velocidad (Fast Ethernet, Gigabit Ethernet y el de 10 Gigabits), redes virtuales, hubs, conmutadores y distintos tipos de medios, tanto de fibra óptica como de cables de cobre (tanto par trenzado como coaxial).

Los estándares de este grupo no reflejan necesariamente lo que se usa en la práctica, aunque a diferencia de otros grupos este suele estar cerca de la realidad.

Tabla 14. Versiones de 802.3

Estándar Ethernet	Fecha	Descripción
Ethernet experimental	1972 (patentado en 1978)	2.94 Mbit/s sobre cable coaxial en topología de bus.
Ethernet II (DIX v2.0)	1982	10 Mbit/s sobre coaxial fino (thinnet) - La trama tiene un campo de tipo de paquete. El protocolo IP usa este formato de trama sobre cualquier medio.
IEEE 802.3	1983	10BASE5 10 Mbit/s sobre coaxial grueso (thicknet). Longitud máxima del segmento 500 metros - Igual que DIX salvo que el campo de Tipo se substituye por la longitud.
802.3a	1985	10BASE2 10 Mbit/s sobre coaxial fino (thinnet o cheapernet). Longitud máxima del segmento 185 metros
802.3b	1985	10BROAD36
802.3c	1985	Especificación de repetidores de 10 Mbit/s
802.3d	1987	FOIRL (Fiber-Optic Inter-Repeater Link) enlace de fibra óptica entre repetidores.
802.3e	1987	1BASE5 o StarLAN
802.3i	1990	10BASE-T 10 Mbit/s sobre par trenzado (UTP). Longitud máxima del segmento 100 metros.
802.3j	1993	10BASE-F 10 Mbit/s sobre fibra óptica. Longitud máxima del segmento 1000 metros.
802.3u	1995	100BASE-TX, 100BASE-T4, 100BASE-FX Fast Ethernet a 100 Mbit/s con auto-negociación de velocidad.
802.3x	1997	Full Duplex (Transmisión y recepción

		simultáneas) y control de flujo.
802.3y	1998	100BASE-T2 100 Mbit/s sobre par trenzado (UTP). Longitud máxima del segmento 100 metros
802.3z	1998	1000BASE-X Ethernet de 1 Gbit/s sobre fibra óptica.
802.3ab	1999	1000BASE-T Ethernet de 1 Gbit/s sobre par trenzado
802.3ac	1998	Extensión de la trama máxima a 1522 bytes (para permitir las "Q-tag") Las Q-tag incluyen información para 802.1Q VLAN y manejan prioridades según el estándar 802.1p.
802.3ad	2000	Agregación de enlaces para enlaces gemelos.
802.3ae	2003	Ethernet a 10 Gbit/s ; 10GBASE-SR, 10GBASE-LR
IEEE 802.3af	2003	Alimentación sobre Ethernet.
802.3ah	2004	Ethernet en el último kilómetro.
802.3ak	2004	10GBASE-CX4 Ethernet a 10 Gbit/s sobre cable bi-axial.
802.3an	en proceso	10GBASE-T Ethernet a 10 Gbit/s sobre par trenzado (UTP)
802.3ap	en proceso	Ethernet de 1 y 10 Gbit/s sobre circuito impreso.
802.3aq	en proceso	10GBASE-LRM Ethernet a 10 Gbit/s sobre fibra óptica multimodo.
802.3ar	en proceso	Gestión de Congestión
802.3as	en proceso	Extensión de la trama

Fuente: Ethernet IEEE 802.3 (pdf) [en línea]. California: Javvin Technologies, Inc., 2002. [Consultado 15 de octubre de 2007]. Disponible en Internet: <http://standards.ieee.org/getieee802/download/802.3.pdf>

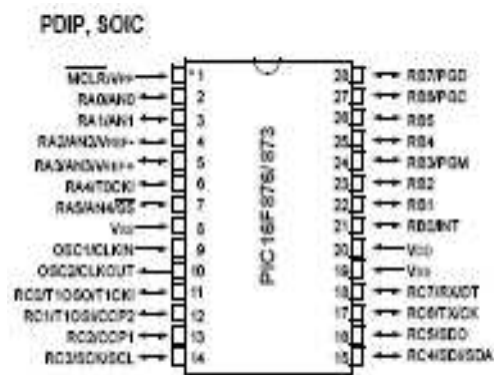
## 2.4. ESPECIFICACIÓN DE INSTRUMENTOS Y EQUIPO DE CONTROL

En este punto se describen las principales características físicas y eléctricas de los dispositivos electrónicos a utilizar en el diseño del proyecto.

2.4.1. Tecnología microcontrolada. Para el desarrollo se selecciono los microcontroladores PIC, como se había mencionado, se escogió la familia 16F87X, la cual es de gama media la cual posee grandes atributos y ventajas frente a otras referencias.

De estos escogimos los microcontroladores PIC16F877A y PIC16F876A que son muy comerciales, y conocidos en el entorno electrónico ver figura 20.

Figura 20. Microcontrolador PIC16F876A



Fuente: PIC 16F877A (pdf) [en línea]. California: Microchip, Inc., 2007. [Consultado 15 de octubre de 2007]. Disponible en Internet: <http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>

Ver anexo 3. Hoja de especificaciones

2.4.2. Conversores RS-232 /Ethernet TCP/IP. Con el propósito de darle una utilidad mas grande al sistema, se hablo anteriormente de que el dispositivo tuviese la capacidad de entregar información a una red especifica dedicada o entregarla a la red de redes, la Internet, con este fin se dio una búsqueda de los diferentes dispositivos que permitieran cumplir con esta función, a continuación se entra en detalle algunos de estos sistemas.

- TelnetPro, Ethernet TCP/IP a RS232

Figura 21. Adaptador Ethernet a RS-232



Fuente: Adaptadores de comunicación [en línea]. Iskra MIS, d.d. 2007. [Consultado 15 de octubre de 2007]. Disponible en Internet: <http://www.iskra-mis.si/index.php?id=380&L=4>

Es un adaptador Ethernet a RS2332, permite conectar un dispositivo RS232 utilizando TCP/IP ver figura 21. Puede ser configurado utilizando un navegador de Internet.

MicroTelnet es un conversor RS232 a protocolo Ethernet. Le permitirá controlar y consultar dispositivos RS232 por medio de Internet abriendo una sesión TCP/IP telnet. Es totalmente configurable a través de un browser de internet. Una gran variedad de parámetros pueden ser configurados para la conexión RS232 e Internet.

MicroTelnet le permitirá hacer accesible cualquier dispositivo con interfase RS232 por medio de Internet. No estará más limitado por la longitud de los cableados. Existen una gran variedad de dispositivos que pueden ser consultados únicamente por RS232. Ahora contróleos por medio de Internet!. Inclusive podrá conectar dispositivos RS232 entre si utilizando dos MicroTelnet.

- RJ45 10BaseT Ethernet interfase
- Transmisión de datos Binarios usando TCP/IP
- Transmisión de datos Binarios usando UDP
- Totalmente compatible RS232 DB-9
- Baud rate desde 50 a 115000 baud
- Parity sin, par o impar
- Flow control: RTS/CTS, XON/XOFF o sin
- Data bit: 7 o 8 bit
- Señales de control disponibles: RTS, CTS, DTR, DMiniSuperRelay, DCD
- Flash ROM actualizable por software por medio de un browser
- Configurable como servidor o cliente

- Conforme a varios standard de internet:
  - 
  - ARP
  - TCP/IP
  - UDP
  - DHCPLink Local / Rendezvous
  - ICMP Ping
  - HTTP
  - Daytime protocol
  - Discard protocol
  - Puertos TCP/IP o UDP pueden ser protegidos utilizando IP y mask hasta 4 IP configurables
  - DHCP
  - Link Local
  - static IP addresses
  - Driver de windows para direccionamiento de COM
  - LED indicador de on-line
  - Alimentación 7.5VDC / 150ma
  - Dimensiones (HxWxD) 0.95 x 2.2 x 2.69 in (2.4 x 5.6 x 6.8 cm)
  
- Net port Ethernet to RS-232 cable adaptor Net Port es un cable adaptador serial a ethernet tipo 'plug-and-play' que permite a cualquier equipo o maquina con un puerto serial, tener acceso a una red o a la Internet en un instante.

El adaptador externo Netport presenta un servidor web integrado, permitiendo a cualquier dispositivo el cual esta conectado, a ser 'contactado' y monitoreado por medio del internet normal. El protocolo TCP/IP esta embebido en el Netport, permitiéndole operar dentro de ambientes sin PC debido a que este presenta un modo de emulación de MODEM, permitiéndole comportarse como uno pero comunicarse por medio de paquetes TCP/IP.

NetPort es un producto el cual puede ser integrado dentro de un sistema con cinismo cambios, una utilidad de configuración es entregada la cual permite rápida y fácil modificación para acomodar a cualquier red o equipo serial.

Los drivers de los puertos virtuales del com son entregados para permitir al dispositivo conectado al NetPort comunicarse con software existente montado sobre TCP/IP (para sistemas operativos windows). De forma alternativa, conexiones de 'sockets directos' pueden ser usados.

Novedades

- 10/100mbps auto sensing rj45 ethernet port



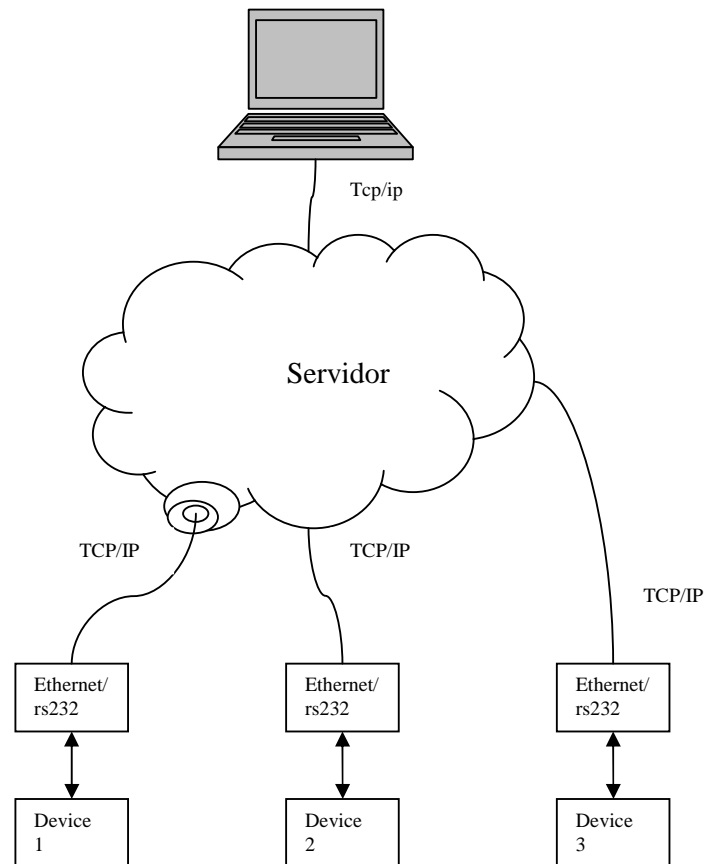
- ethernet activity and status led's
- page web server
- 234kbps maximum serial throughput
- moulded 9-way d-type serial cable (standard)
- serial & power status leds
- 128bit rijndael aes encryption (lan03/4)
- can be supplied with psu
- power pass-through to/from serial equipment (lan06only)
- power consumption 240ma @ 7.5vdc, 75ma @ 24vdc
- configurable security options
- can be customised for oem use

## 2.5. ESPECIFICACIONES DE MONTAJE

Para definir esta parte, debemos conocer el medio en donde generalmente el equipo será instalado, para eso se mostrara, como el equipo interactúa con este ambiente y con otro tipo de equipos que sean necesarios en el desempeño de este.

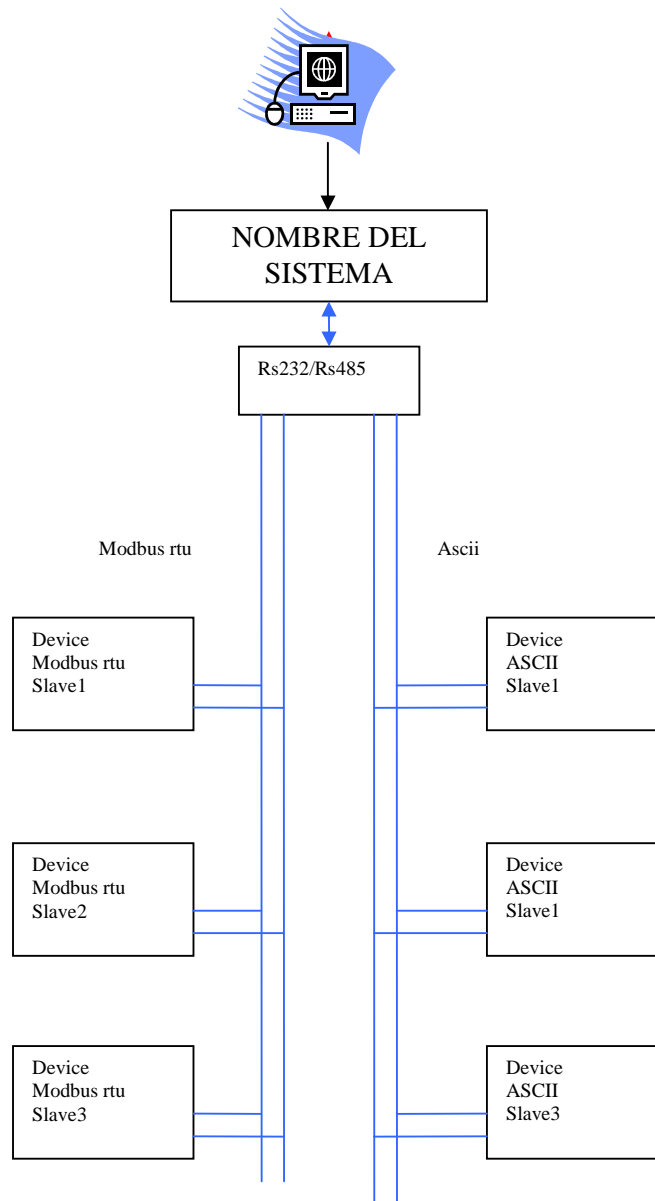
2.5.1. Ubicación del equipo. El sistema como sabemos, se comunicara con un computador vía RS-232, ya si se desea hacer monitoreo externo del sistema, se le acopla un conversor RS-232/ethernet(TCP-IP) con el propósito de poder acceder al dispositivo vía Internet, o una red dedicada., en ambos casos se requiere que el sistema este a una distancia prudente del sistema, según el caso , los estándares indican un a distancia máxima en la cual el sistema podrá transmitir o recibir sin inconvenientes, en este caso las normas estipuladas en los estándares EIA/TIA-232-E y el IEEE 802.3 (ethernet), de los cuales ya se han definido sus características.

Figura 22. Ubicación del equipo



En la figura 22 se muestra como sería de forma global la interconexión entre equipos si se desea hacer un monitoreo desde una fuente externa (usando el conversor ethernet) de cada tarjeta.

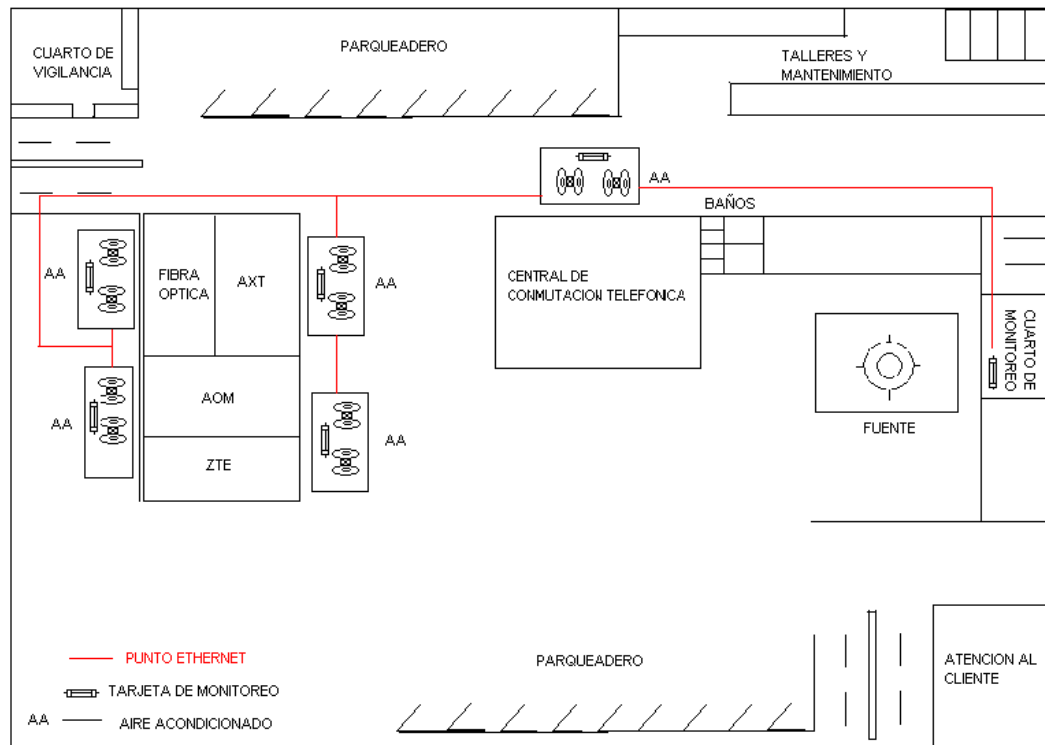
Figura 23. Conexión de una sola tarjeta al PC



En la figura 23 se muestra el sistema usando solamente una tarjeta conectada aun cpu por medio de RS-232, a la cual se le conectan dispositivos en red por medio de RS-485. Como ya se especifico, el RS-485, permite manejar grandes distancias entre equipos esclavos al maestro, en este caso seria la tarjeta. El bloque que aparece como RS-232/RS-485 no hace parte del diseño pero es necesario si se desea conectar en red a varios equipos, ya sean equipos modbus, o equipos ASCII.

## 2.6. DIAGRAMA DE LOCALIZACIÓN DE INSTRUMENTOS

Figura 24. Plano de EMCALI ubicación instrumentos



## 2.7. LISTADO Y COTIZACIÓN DE INSTRUMENTOS Y MATERIALES.

Tabla 15. Listado de materiales

<b>LISTADO DE MATERIALES ELECTRONICOS UTILIZADOS</b>			
<b>ELEMENTO</b>	<b>CANTIDAD</b>	<b>VALOR UNITARIO</b>	<b>VALOR TOTAL</b>
PIC16F877	1	20000,00	20000
PIC16F876	2		0
MAX232	3	4000,00	12000
CONECTOR DB-9	3	600,00	1800
CARCASAS DB-9	3	500,00	1500
LM7805	1	1500,00	1500
MOC3011	8	1600,00	12800
R-1K	10	100,00	1000
TRANSF-REDUCTOR	1		0
74LS04	5	1200,00	6000
CONDENSADORES	2	200,00	400
DIODO-LED	1	200,00	200
PUENTE RECTIFICADOR 3A	1	800,00	800
DIODO RECTIFICADOR	1	100,00	100
PLAQUETA DE IMPRESIÓN 20*20	1	4000,00	4000
LICENCIAS	1	550000,00	550000
MONTAJE DE TARJETAS	1	250000,00	250000
NODO FUNCIONANDO	1	1500000,00	1500000
SWITCH	1		0
CONVERSION 232/ETHERNET	1	320000,00	320000
TOTAL			2682100

A partir de lo esbozado en el capítulo anterior, se puede concluir que el sistema trazado posee muchas partes a diseñar, lo cual indica que será importante que el proyecto pueda manejar todos los módulos sin presentar problemas y limitaciones, ya conocido lo anterior se continua con la representación real de los diseños en bloque de cada modulo pretendido, y también se muestra como quedará constituido el sistema en su final.

### 3. INGENIERÍA DE DETALLE: PROTOTIPO FINAL

#### 3.1. DIAGRAMAS SECUENCIALES, DIAGRAMAS DE FLUJO DE VARIABLES DE COMUNICACIONES.

A continuación se mostrara como se ha desarrollado el proyecto, cada una de sus partes y como interactúa cada una de ellas, se mostrara a partir de los sistemas básicos a los más generales.

##### 3.1.1. Comunicaciones

- Comunicaciones internas
- El sistema posee una serie de métodos por los cuales sus subsistemas intercambian información
- Comunicación a cuatro bits bidireccional
- Como dice el nombre este sistema, envía información a cuatro bits, simplemente coge un byte y lo parte para después enviar 2 paquetes de 4 bits, esto se hizo para poder economizar el uso de puertos del microcontrolador ver figura 25.

Figura 25. Comunicación bidireccional

```
#int_EXT
EXT_isr()
{

    output_low(pin_b3);
    switch (i) {
        case 0: set_tris_a(0xcf);
                dato1 = input_a();
                i++;
                break;
        case 1: set_tris_a(0xcf);
                dato2 = input_a();
```

```

aux.data1=dato1.data1;
aux.data2 = dato2.data1;
dato[j]=aux;

if (j==dato[0]-48)
    enabledata=1;
    for (i=0; i<=j-1; i++)
    {
        buffer[i]=dato[i+1]-48;
    }
    j=0;
}
else{ j++;}

output_high(pin_b3);
break;

default:

    break; }

}

```

Este código se ejecuta cuando ocurre una interrupción externa proveniente del modulo que envía la información, con cada interrupción se cogen cuatro bits, y por medio del código se arma el mensaje, como se muestra arriba, el código que se muestra arriba es la parte de recepción ahora mostraremos la transmisión ver figura 26.

Figura 26. Comunicación bidireccional 4 bits

```

set_tris_a(0);
for (j=0; j<=5; j++)
{
    output_low (pin_B1);
    output_a(0);
    SWAP(error);
    output_a(error);
    output_high (pin_B1);
    delay_us(100);
    output_low (pin_B1);
    delay_us(100);
    SWAP(error);
    output_a(error);
    output_high (pin_B1);
}

```

```

set_timer2(0);
delay_ms(10);
}

```

Esta es la estructura de la transmisión de la información, se dice eso, por que el código sufre algunas diferencias en los diferentes módulos.

- Comunicaciones externas. Se mostrara como fue el diseño de los protocolos modbus rtu master y el ASCII.

- Modbus rtu master. el protocolo modbus posee diferentes tramas las cuales se usan para adquirir alguna clase de información deseada, en el diseño se han implementado estas funciones de la siguiente manera ver figura 27.

Figura 27. Funciones modbus

```

void readcoilstatus(int SlaveAddress,int End_Coil)
{
    //!seis caracteres de respuesta
    buffer[0]=SlaveAddress;
    buffer[1]=0x01;
    buffer[2]=0;
    buffer[3]=End_Coil;
    buffer[4]=0;
    buffer[5]=1;
    size=6;
    output_high(pin_b7);
}

void ReadHoldingRegister(int SlaveAddress,int Start_Coil,int End_Coil)
{
    //!siete caracteres de respuesta
    buffer[0]=SlaveAddress;
    buffer[1]=0x03;
    buffer[2]=Start_Coil;
    buffer[3]=End_Coil;
    buffer[4]=0;
    buffer[5]=1;
    size=6;
    output_high(pin_b6);
}

void ReadInputRegisters(int SlaveAddress,int Start_Coil,int End_Coil,int Tcoil)
{
    buffer[0]=SlaveAddress;
    buffer[1]=0x04;

```



```

    buffer[2]=Start_Coil;
    buffer[3]=End_Coil;
    buffer[4]=0;
    buffer[5]=Tcoil;
    size=6;
    output_high(pin_b5);
}

```

```

void ForceSingleCoil(int SlaveAddress,int End_Coil,int ForceDataHigh)
{
    //!ocho caracteres de respuesta ,igual al mensaje si no hay error
    buffer[0]=SlaveAddress;
    buffer[1]=0x05;
    buffer[2]=0;
    buffer[3]=End_Coil;
    buffer[4]=ForceDataHigh;
    buffer[5]=0;
    size=6;
    output_high(pin_b4);
}

```

```

void PresetSingleRegister(int SlaveAddress,int AddressHigh,int Addresslow,int
DataHigh,int DataLow)
{
    //!ocho caracteres de respuesta,igual al mensaje si no hay error
    buffer[0]=SlaveAddress;
    buffer[1]=0x06;
    buffer[2]=AddressHigh;
    buffer[3]=Addresslow;
    buffer[4]=DataHigh;
    buffer[5]=DataLow;
    size=6;
    output_high(pin_b3);
}

```

```

void ForceMultipleCoils(int SlaveAddress,int CoilAddressHigh,int
CoilAddressLow,int CoilsHighOrder,int CoilsLowOrder,int ByteCount,int
ForceDataHigh)
{
    buffer[0]=SlaveAddress;
    buffer[1]=0x0F;
    buffer[2]=CoilAddressHigh;
    buffer[3]=CoilAddressLow;
    buffer[4]=CoilsHighOrder;
    buffer[5]=CoilsLowOrder;
    buffer[6]=ByteCount;
    buffer[7]=ForceDataHigh;
    size=8;
}

```

```

void PresetMultipleRegisters(int SlaveAddress,int StartingRegister,int
TRegisters,int DataHigh,int DataLow )
{

    //!revisar que es multiple

    buffer[0]=SlaveAddress;
    buffer[1]=0x10;
    buffer[2]=StartingRegister;
    buffer[3]=TRegisters;
    buffer[4]=0;
    buffer[5]=1;
    buffer[6]=DataHigh;
    buffer[7]=DataLow;
    size=8;
}

```

Por medio del código se han creado funciones que reciben unos parámetros los cuales entregan la información a enviar.

A continuación se muestra el código que decide que tipo de función modbus se ha enviado ver figura 28.

Figura 28. Identificador de función modbus

```

void readdatapp()
{
    switch (buffer[1]) {
        case 1:
            readcoilstatus(buffer[0],buffer[2]);
            break;
        case 3:
            ReadHoldingRegister(buffer[0],buffer[2],buffer[3]);
            break;
        case 4:
            ReadInputRegisters(buffer[0],buffer[2],buffer[3],buffer[4]);
            break;
        case 5:
            ForceSingleCoil(buffer[0],buffer[2],buffer[3]);
            break;
        case 6:
            PresetSingleRegister(buffer[0],buffer[2],buffer[3],buffer[4],buffer[5]);
            break;
        case 16:
            break;
        default:break; }
}

```

Cuando se ha confirmado que trama modbus se usara y se tiene la información a enviar, entonces al mensaje se le calcula lo que se conoce como CRC (código de redundancia cíclica) el cual se usa por el receptor para poder verificar el estado de la información ver figura 29.

Figura 29. Generador de CRC

```
void GenerateCRC(int c)
{
    lastbit=0;
    crc=0xffff;
    for (i=0; i<c; i++)
    {
        thisbyte= buffer[i];
        crc = crc^thisbyte;
        for (shift=1; shift<=8; shift++)
        {
            lastbit = crc & 1;
            crc = (crc >> 1) & 0x7fff;
            if (lastbit==1)
            {
                crc = crc^0xA001;
            }
        }
    }
    CRCHigh=(crc>>8)&0xff;
    CRCLow=crc&0xff;
}
```

Se ha vuelto en forma de función, solo recibe un parámetro que es el tamaño del mensaje, con este el calcula los 2 valores del crc, crclow y crchigh, los cuales serán transmitidos en el mensaje.

Al culminar esta parte el mensaje esta listo para ser enviado ver figura 30

Figura 30. Transmisor de datos.

```
void transdata()
{
    for (i=0;i<=size;++i)
    {
        putc(buffer[i]);
        delay_ms(5); //! el retardo entre caracteres máximo debe ser aprox. 6.875 ms
    }
    putc(CRCLow);
}
```

```

    delay_ms(5);
    putc(CRCHigh);
    i=0;
}

```

El mensaje es enviado según las especificaciones del modbus acerca de tiempos entre caracteres enviados, como se muestra en la figura 31.

Ahora el sistema espera a que el esclavo modbus responda a su llamado.

Figura 31. Recepción de datos

```

#int_RDA
RDA_isr()
{
    buffer[i]=fgetc();
    i++;
    delay_ms(100);
    disable=1;
}

```

Los datos que llegan son guardados, también se acciona un retardo el cual es para determinar los mensajes que llegan, según dice el protocolo modbus rtu, el maestro debe esperar un tiempo no superior a 1.5 veces el tiempo en que tarda de transmitir un carácter según la velocidad de transmisión, esto se explico con mayor detalle en la sección de know how.

Figura 32. Reconocimiento de información proveniente de la red.

```

if (disable==1){
    disable=0;
    ans_data();
}

```

Este código es creado para activarse solo cuando a transcurrido el time \_out entre tramas, entonces el sistema procede a evaluar el mensaje ver figura 32.

```

void ans_data()
{
    if( (i<4)||i>30){ clrbuffer();wrong_data();}
    else{
        size1=i;
        GenerateCRC(i-2);
    }
}

```

```

if((CRCHigh==buffer[size1-1])&&(CRCLow==buffer[size1])){
    set_tris_a(0);
    output_low (pin_B1);
    output_a(0);
    SWAP(size1);
    output_a(size1);
    output_high (pin_B1);
    delay_us(100);
    output_low (pin_B1);
    delay_us(100);
    SWAP(size1);
    output_a(size1);
    output_high (pin_B1);
    delay_ms(10);
    for (j=0; j<=size1; j++)
    {
        output_low (pin_B1);
        output_a(0);
        SWAP(buffer[j]);
        output_a(buffer[j]);
        output_high (pin_B1);
        delay_us(100);
        output_low (pin_B1);
        delay_us(100);
        SWAP(buffer[j]);
        output_a(buffer[j]);
        output_high (pin_B1);
        delay_ms(10);
    }
}
else{wrong_data();}
i=0;
size1=0;
}
}

```

Este código se encarga de reconocer el mensaje, preguntado si el mensaje tiene un tamaño razonable, y también se encarga de calcular el CRC del mensaje entrante, para luego compararlo con el CRC que llevo, si es correcto el sistema procederá a enviar la información extraída, sino enviar a una trama de error al sistema.

```

void wrong_data()
{ set_tris_a(0);
  for (j=0; j<=5; j++)
  {
    output_low (pin_B1);
    output_a(0);

```

```

    SWAP(error);
    output_a(error);
    output_high (pin_B1);
    delay_us(100);
    output_low (pin_B1);
    delay_us(100);
    SWAP(error);
    output_a(error);
    output_high (pin_B1);
    set_timer2(0);
    delay_ms(10);
}

```

Este es el código que configura al micro controlador y se encarga de manejar las tareas del modbus, transmisión y recepción ver figura 33.

Figura 33. Código principal

```

void main()
{
    output_b(0);
    SET_TRIS_B(0x01);
    SET_TRIS_C(0x80);
    setup_adc_ports(NO_ANALOGS);
    setup_adc(ADC_OFF);
    setup_spi(FALSE);
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DIV_BY_16,85,11);//! timeout entre tramas debe ser de
    aprox 16.041ms (2400 bauds)
    setup_comparator(NC_NC_NC_NC); //!, se desborda cada 15.1ms
    setup_vref(FALSE);

    enable_interrupts(INT_EXT);
    enable_interrupts(INT_RDA);
    output_low(pin_b1);
    enable_interrupts(GLOBAL);
    while(1)
    {
        if (enabledata==1)
        {
            enabledata=0;

            readdatapp();
            GenerateCRC(size);
            transdata();

```

```

if (disable==1){
    disable=0;
    ans_data();
}
}
}

```

Este sistema es mas sencillo que el anterior, solo debemos enviar de forma serial el dato que llega paralelo, y viceversa.

Figura 34. Protocolo ASCII

```

int i=0,dato=0;
#int_EXT
EXT_isr()
{
    set_tris_b(0x01);
    output_low(pin_b3);
    switch (i) {
        case 0: set_tris_a(1);
                dato1 = input_a();
                i++;
                set_tris_a(0);
                break;
        case 1: set_tris_a(1);
                dato2 = input_a();
                i=0;
                aux.data1=dato1.data1;
                aux.data2 = dato2.data1;
                dato=aux;
                output_high(pin_b3);
                putc(dato);
                set_tris_a(0);
                break;

        default:
            break; }
    }

#int_RDA
RDA_isr()
{
    set_tris_a(1);
    dato=fgetc();
    set_tris_b(0x01);
    output_low(pin_b1);
}

```

```

output_a(dato);
output_high(pin_b1);
delay_us(100);
output_low(pin_b1);
delay_us(100);
SWAP(dato);
output_a(dato);
output_high(pin_b1);
set_tris_a(0);
}

```

Como se ve en la figura 34, el sistema se quede esperando a que ocurra una interrupción externa o serial, para hacer el paso de información, utiliza módulos de código ya vistos en el sistema anterior.

- Sistema principal. El sistema principal, es como fue mostrado en la parte de diseño conceptual, una integración de tareas, las cuales son administradas de acuerdo a la información proveniente de PC, queriendo decir con esto que este sistema es el que se debe ejecutar, y como lo hará. Para esto se genero un sistema que establece prioridad a la hora de poder ejecutarse cualquier programa. A continuación mostraremos como es la administración de tareas y como se ejecutan.

Figura 35. Recepción de datos seriales.

Los datos que son enviados que llegan por el serial del sistema, son datos enviados entre caracteres asciii, los cuales son usados para identificar si la trama es correcta o no.

```

#int_RDA
RDA_isr()
{
    bufferss[i]=fgetc();

    if (i>=8)
    {
        if((bufferss[0]==65)&&(bufferss[1]==65)&&(bufferss[i-1]==90)&&(bufferss[i]==90))
        {
            if(bufferss[2]==89){beginascii=1;}
            if(bufferss[3]==89){beginmodbus=1;}
            if(bufferss[4]==89){beginanag=1;}

            i=0;
        }
    }
}

```



```

    }
  }
  i++;
}

```

El sistema va preguntado por los caracteres que van llegando hasta que puede identificar los datos de la trama que son esperados, entonces el sistema acciona la tareas que el PC a ordenado y las que no ver figura 35.

Figura 36. Código principal

En esta parte el sistema evalúa las tareas que han sido asignadas para hacer, y las ejecuta.

```

void main()
{
    inicio:
    if(inicia==0){zero();delay_ms(800);output_high (pin_C4);goto inicio;}
    if(param==0){set();delay_ms(800);output_high (pin_C5);goto inicio;}

    while(1)
    {
        if (beginascii==1){senddatascii();}
        if
        ((endpsp==1)&&(beginascii==0)&&(beginmodbus==1)){senddatamodbus();}
        if
        ((endpsp==1)&&(beginascii==0)&&(beginmodbus==0)&&(beginanag==1)){tarea
        _dato_digital();tarea_anag_digi();}
        if
        ((endpsp==1)&&(beginascii==0)&&(beginmodbus==0)&&(beginanag==0)&&(be
        gintrans==1)){transmision();}

    }
}

```

En esta parte también se utiliza un pequeño código de inicialización y configuración del sistema ver figura 36, esto es para poder garantizar un correcto funcionamiento del sistema, el usuario lo puede observar, que el sistema posee 2 salidas la cuales indican si estas fases han sido cumplidas con éxito. La figura 37 muestra el código que se uso para estas configuraciones.

Figura 37. Configuración del micro controlador

```
void set()
{ SET_TRIS_E(0x00);
  SET_TRIS_B(0xF9);
  SET_TRIS_C(0xFF);

  setup_adc_ports(AN0_AN1_AN2_AN3_AN4_AN5);
  setup_adc(ADC_CLOCK_INTERNAL);
  setup_psp(PSP_DISABLED);
  setup_spi(FALSE);
  setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
  setup_timer_1(T1_DISABLED);
  setup_timer_2(T2_DISABLED,0,1);
  setup_comparator(NC_NC_NC_NC);
  setup_vref(FALSE);
  ext_int_edge( L_TO_H );
  enable_interrupts(INT_EXT);
  enable_interrupts(INT_RDA);
  enable_interrupts(GLOBAL);

  param=1;
}
```

Figura 38. Inicialización de variables, puertos, etc.

```
void zero()
{
for (i=0;i<=40;++i)
{
  bufferout[i]='0';
}
beginascii=0;
beginmodbus=0;
begintrans=0;

output_a(0);
output_b(0);

output_d(0);
output_e(0);

i=0;
inicia=1;
trama=0;
}
```

Para esta parte se usa el modulo de transmisión en paralelo que se había visto anteriormente.

Figura 39. Sistema de transmisión de datos ASCII.

```
void senddatascii()
{
    j=0;
    output_high (pin_e0);
    SET_TRIS_B(0x01);
    SET_TRIS_D( 0xF0 );
    output_d(0);
    output_LOW (pin_B1);
    for (i=0;i<=bufferss[5]-49;++i)
    {
        output_low (pin_B1);
        port_b=bufferss[i+7];
        data[i]=bufferss[i+7];
        output_d(port_b.data1);
        output_high (pin_B1);
        delay_us(100);
        output_low (pin_B1);
        delay_us(100);
        output_d(port_b.data2);
        output_high (pin_B1);
        delay_ms(60);
    }
    beginascii=0;
    SET_TRIS_D( 0x0F );
    i=0;
}
```

Como se había dicho el sistema funciona de forma similar a los otros, solo que se hacen unas modificaciones para el manejo del puerto ver figura 39 (se recuerda que este microcontrolador utiliza solo un puerto para distribuir la información a los otros microcontroladores) pero en lo demás es igual.

Funciona de la misma manera que el sistema anterior.

Figura 40. Sistema de transmisión de datos MODBUS RTU.

```
void senddatamodbus()
{
    j=0;
    output_high (pin_e1);
    SET_TRIS_D( 0x0F );
    output_d(0);
    output_LOW (pin_B2);
    for (i=0;i<=bufferss[6]-49;++i)
    {

        output_low (pin_B2);
        output_d(0);
        port_b=bufferss[i+7+bufferss[5]-48];
        data[i]=bufferss[i+7+bufferss[5]-48];
        SWAP(data[i]);
        output_d(data[i]);
        output_high (pin_B2);
        delay_us(100);
        output_low (pin_B2);
        delay_us(100);
        SWAP(data[i]);
        output_d(data[i]);
        output_high (pin_B2);
        delay_ms(60);
    }
    beginascii=0;
    SET_TRIS_D( 0xF0 );
    beginmodbus=0;
    listo=1;
    endpsp=0;
    i=0;
}
```

Este sistema funciona de la misma manera que los otros sistemas que se usaron para poder recoger información a cuatro bits, con la diferencia de que este sistema es usado tanto para recoger datos ASCII, como datos modbus.

Figura 41. Sistema de recepción de datos paralelos.

```
#int_EXT
EXT_isr()
{
```

```

if (j==2){j=0;}

output_high (pin_e2);
data[j]=input_d();
if (j==1)
{
    dato1=data[0];
    dato2=data[1];
    if(listo==1)
    {
        aux.data1=dato1.data2;
        aux.data2=dato2.data2;
    }
    else
    {
        aux.data1=dato1.data1;
        aux.data2=dato2.data1;
    }

    bufferout[i+trama]=aux;
    if (i==0){datalenght= bufferout[0+trama]-48;}
    i++;
}
j++;
if (i==datalenght){output_d(0);endpsp=1;trama=trama+i;i=0;}
}

```

Este sistema reconoce y separa la información proveniente de los otros dos microcontroladores, esto se hace mirando que el sistema ya término su tarea, así se sabe de quien es la información ve figura 41.

- Recolección de datos digitales y analógicos. Cuando el sistema a terminado de enviar y leer información de los microcontroladores externos, esta tarea se ejecuta para obtener los datos de los puerto analógicos y digitales.

El microcontrolador PIC16F877A, como ya se había mencionado, posee una gran periferia la cual nos permite hacer diversas tareas, entre esas tareas se encuentran sus puertos analógicos, los cuales resultan ser muy eficientes y fáciles de utilizar ver figura 42.

Figura 42. Recolección datos análogos.

```

void tarea_anag_digi()
{

```

```

output_high (pin_e2);
for (j=0;j<=5;++j)
{
    set_adc_channel(j);
    delay_us(10);
    bufferout[trama+j]= read_adc(ADC_START_AND_READ );
}

beginanag=0;
begintrans=1;
}

```

El código obtiene su información de 6 canales análogos, que son la totalidad de canales que posee el microcontrolador, cada uno se lee de forma individual, el sistema no permite obtener la información de todos a la vez.

Figura 43. Recolección de datos digitales.

```

void tarea_dato_digital()
{
    bufferout[trama]=input_state(pin_b3)+48;
    Trama++;
    bufferout[trama]=input_state(pin_b4)+48;
    Trama++;
    bufferout[trama]=input_state(pin_b5)+48;
    Trama++;
    bufferout[trama]=input_state(pin_b6)+48;
    Trama++;
    bufferout[trama]=input_state(pin_b7)+48;
    Trama++;
    bufferout[trama]=input_state(pin_C0)+48;
    Trama++;
    bufferout[trama]=input_state(pin_C1)+48;
    Trama++;
    bufferout[trama]=input_state(pin_C2)+48;
    Trama++;
}

```

El compilador permite preguntar por el estado de cada pin, esta información es guardada como un 1,0 ASCII ver figura 43 para la captura de la información digital.

Esta parte del sistema se ejecuta cuando el ya todas las tareas pedidas por el p can sido ejecutadas.

Figura 44. Transmisión de datos al PC

```
void transmisión()
{
    putc('A');    //!transmision de datos a la red tcp/ip
    putc('A');    //!trama
    putc(trama);
    for (i=0;i<=39;++i)
    {
        putc(bufferout[i]);
    }
    putc('Z');
    putc('Z');
}
```

Como se observa en la figura 44, la información es encapsulada en una trama de respuesta para que el PC pueda identificar cada una de sus partes con facilidad.

- Software de diseño. El software que se utilizara para el manejo de la información obtenida por el sistema electrónico se hizo de tal forma que fuera de facil configuración acceso, y manejo, como opera el software será explicado en el manual de usuario que viene en los anexos.
- Modulo configuración En este modulo el operario podrá hacer configuraciones, con respecto a la red, la generación de alarmas, tiempos de recolección y generación de información.

Esta función se ejecuta cuando es accionado uno de los botones Della pantalla, esta me permite verificar el usuario a ingresado todos los campos necesarios para que el software opere ver figura 45.

Figura 45. Aceptación de información

```
Private Sub Aceptar_Click()
If direccion = " " Or adinic = " " Or npoints = " " Then
    MsgBox ("Configuración del modbus incompleta")
Else
    If comp1 = " " Or comp2 = " " Or comp3 = " " Or comp4 = " " Or comp5 = " "
Then
        MsgBox ("Configuración de datos análogos incompleta")
    Else
        If dataascii = " " Then
```

```

        MsgBox ("Configuración del ASCII incompleta")
    Else
        If ipaddress = " " Or puertoremoto = " " Or puertolocal = " " Then
            MsgBox ("Configuración de la red incompleta")
        Else
            If tiemporefresco = " " Or tarchivos = " " Then
                MsgBox ("Configuración de temporizadores incompleta")
            Else
                Load Principal
                Parametrizacion.Hide
                Principal.Show
                enable = True
            End If
        End If
    End If
End If
End Sub

```

Esta función lo que hace es simplemente guardar los datos obtenidos en la pantalla ver figura 46.

Figura 46. Recoge datos

```

Private Sub Aplicar_Click()

    dirección = Address.Text
    adinic = Starting_address.Text
    npoints = N_points.Text

    comp1 = compara1.Text
    comp2 = compara2.Text
    comp3 = compara3.Text
    comp4 = compara4.Text
    comp5 = compara5.Text

    dataascii = trama_ascii.Text

    tiemporefresco = Trefresco.Text
    tarchivos = Generaarchivo.Text

    ipaddress = IP_address.Text
    ipaddress1 = IP_address1.Text
    ipaddress2 = IP_address2.Text
    ipaddress3 = IP_address3.Text

```



```
ipaddress4 = IP_address4.Text  
  
puertoremoto = Remote_port.Text  
puertolocal = Local_port.Text
```

```
End Sub
```

Esta función solo borra la ingresada por el usuario, si desea ingresar nueva información ver figura 47.

Figura 47. Cancela información

```
Private Sub Command1_Click()
```

```
Address.Text = " "  
Starting_address.Text = " "  
N_points.Text = " "
```

```
compara1.Text = " "  
compara2.Text = " "  
compara3.Text = " "  
compara4.Text = " "  
compara5.Text = " "
```

```
trama_ascii.Text = " "
```

```
Trefresco.Text = " "  
Generaarchivo.Text = " "
```

```
IP_address.Text = " "  
Remote_port.Text = " "  
Local_port.Text = " "  
End Sub
```

Estas 2 funciones se encargan de cambiar el método físico por el cual se desea enviar la información ver figura 48.

Figura 48. Ethernet o serial

```
Private Sub Option1_Click()
```

```

serial = True
ethernet = False
Option2.Enabled = False
End Sub

```

```

Private Sub Option2_Click()
ethernet = True
serial = False
Option1.Enabled = False
End Sub

```

Por medio de esta función el sistema configura los datos iniciales que se requieren para poder hacer la configuración de la trama MODBUS RTU ver figura 49.

Figura 49. Trama Modbus a enviar

```

Private Sub tipotrama_Change()

Select Case tipotrama.List
    Case "Lea estado de las bobinas"
        tramatipe = 1
        tamaño = 8
        función = 1
    Case "Lea estado de las entradas"
        tramatipe = 2
        tamaño = 8
    Case "Lea registros de entrada"
        tramatipe = 3
        tamaño = 8
    Case "Lea estado de los registros"
        tramatipe = 4
        tamaño = 8
    Case "Cambiar estado de bobina"
        tramatipe = 8
    Case "Cambiar estado de registro"
        tramatipe = 6
    Case "Cambiar multiple bobinas"
        tramatipe = 15
End Select

End Sub

```

- Modulo Monitoreo. En este modulo se puede observar el estado de la información que llega del sistema electrónico. Esta función lo que hace es recoger la información obtenida en la parte de configuración, para poder configurar lo que el operador desea ver figura 50.

Figura 50. Función carga

```
Private Sub Form_Load()

With udpa
.RemoteHost = ipaddress
.RemotePort = puertoremoto
.Bind puertolocal
End With

Timer1.Interval = tiemporefresco * 1000
Timer1.Enabled = True

Timer2.Interval = tarchivos * 1000
Timer2.Enabled = True

porthmi.InBufferCount = 0
porthmi.OutBufferCount = 0
porthmi.RThreshold = 46
porthmi.InputLen = 46
DoEvents

End Sub
```

Esta función solo se encarga de cambiar de ventana, pasa de la ventana actual que monitoreo a parametrización, esto se hace para dar la posibilidad al operador de hacer modificación si se desea ver figura 51.

Figura 51. Función para cambiar ventanas

```
Private Sub Param2_Click()

Operacion.Hide
Load Parametrizacion
Parametrizacion.Show
End Sub
```

Esta función se encarga de la recepción de datos seriales provenientes de la red (en caso de que este conectado por el conversor rs232/Ethernet) o el dispositivo, visual Basic posee módulos seriales los cuales permiten con mucha facilidad esta tarea, aquí mismo se hace la separación de la información y su aplicación ver figura 52.

Figura 52. Función reconocedora de datos (serial Rs232)

```
Private Sub porthmi_OnComm()  
  
Select Case porthmi.CommEvent  
    ' Controlar cada evento o error escribiendo  
    ' código en cada instrucción Case  
  
    ' Errores  
    Case comBreak ' Se ha recibido una interrupción.  
    Case comEventFrame ' Error de trama  
    Case comEventOverrun ' Datos perdidos.  
    Case comEventRxOver ' Desbordamiento del búfer  
        ' de recepción.  
    Case comEventRxParity ' Error de paridad.  
    Case comEventTxFull ' Búfer de transmisión lleno.  
    Case comEventDCB ' Error inesperado al recuperar DCB.  
  
    ' Eventos  
    Case comEvCD ' Cambio en la línea CD.  
    Case comEvCTS ' Cambio en la línea CTS.  
    Case comEvDSR ' Cambio en la línea DSR.  
    Case comEvRing ' Cambio en el indicador de  
        ' llamadas.  
    Case comEvReceive ' Recibido nº SThreshold de  
        ' caracteres.  
    Timer1.Enabled = False  
    Timer2.Enabled = False  
    trama = porthmi.Input  
    textrepcion.Caption = trama  
    If Mid(trama, 1, 2) = "AA" And Mid(trama, 44, 2) = "ZZ" Then  
        ndatosascii = Mid(trama, 4, 1)  
        ndatosmodbus = Mid(trama, 4 + ndatosascii, 1)  
  
        datosascii = Mid(trama, 5, ndatosascii)  
        datosmodbus = Mid(trama, 5 + ndatosascii, datosmodbus)  
  
        Modbustext.Text = datosmodbus  
        ASCIItext.Text = datosascii  
  
        For i = 0 To 7  
            If Mid(trama, ndatosmodbus + ndatosascii + 3 + i, 1) = "1" Then  
                objeto(i).BackColor = &HFF&  
            Else  
                objeto(i).BackColor = &HFF00&  
            End If  
        Next i  
    End If  
End Sub
```

```

Next i

anag1 = Asc(Mid(trama, ndatosmodbus + ndatosascii + 3 + 9, 1)) *
100 / 256
anag2 = Asc(Mid(trama, ndatosmodbus + ndatosascii + 3 + 10, 1))
* 100 / 256
anag3 = Asc(Mid(trama, ndatosmodbus + ndatosascii + 3 + 11, 1))
* 100 / 256
anag4 = Asc(Mid(trama, ndatosmodbus + ndatosascii + 3 + 12, 1))
* 100 / 256
anag5 = Asc(Mid(trama, ndatosmodbus + ndatosascii + 3 + 13, 1))
* 100 / 256
anag6 = Asc(Mid(trama, ndatosmodbus + ndatosascii + 3 + 14, 1))
* 100 / 256

For i = 8 To 12
    If anag1 = Asc(comp1) Then
        objeto(i).BackColor = &HFF&
    Else
        objeto(i).BackColor = &HFF00&
    End If

    If anag2 = Asc(comp2) Then
        objeto(i).BackColor = &HFF&
    Else
        objeto(i).BackColor = &HFF00&
    End If

    If anag3 = Asc(comp3) Then
        objeto(i).BackColor = &HFF&
    Else
        objeto(i).BackColor = &HFF00&
    End If

    If anag4 = Asc(comp4) Then
        objeto(i).BackColor = &HFF&
    Else
        objeto(i).BackColor = &HFF00&
    End If

    If anag5 = Asc(comp5) Then
        objeto(i).BackColor = &HFF&
    Else
        objeto(i).BackColor = &HFF00&
    End If
Next i

```

```

        Nivel1.Caption = anag1
        Nivel2.Caption = anag2
        Nivel3.Caption = anag3
        Nivel4.Caption = anag4
        Nivel5.Caption = anag5
        Nivel6.Caption = anag6

    Else
        MsgBox ("Error de trama de llegada desde el PLC")
    End If
    textrepcion.Caption = textrepcion.Caption + trama

    Timer1.Enabled = True
    Timer2.Enabled = True
    Case comEvSend ' Hay un SThreshold
        ' caracteres en el búfer
        ' de transmisión.
    Case comEvEOF ' Se ha encontrado un carácter
        ' EOF en la entrada.
End Select
End Sub

```

Esta función opera de forma similar a la anterior solo que esta maneja es controlada por el puerto Ethernet y es controlada por el protocolo udp, visual basic posee un modulo de comunicación de este tipo, lo cual facilita el manejo de este en el software ver figura 53.

Figura 53. Función reconocedora de datos ( UDP)

```

Private Sub udpa_DataArrival _
    (ByVal bytesTotal As Long)

    Dim strData As String

    udpa.GetData strData
    textrepcion.Caption = strData
    If Mid(strData, 1, 5) = "AA" And Mid(strData, 44, 2) = "ZZ" Then
        ndatosascii = Mid(strData, 4, 1)
        ndatosmodbus = Mid(strData, 4 + ndatosascii, 1)

        datosascii = Mid(strData, 5, ndatosascii)
        datosmodbus = Mid(strData, 5 + ndatosascii, ndatosmodbus)

        Modbustext.Text = datosmodbus
        ASCIItext.Text = datosascii

        For i = 0 To 7
            If Mid(strData, ndatosmodbus + ndatosascii + 3 + i, 1) = "1" Then

```

```

        objeto(i).BackColor = &HFF&
    Else
        objeto(i).BackColor = &HFF00&
    End If
Next i

anag1 = Asc(Mid(trama, ndatosmodbus + ndatosascii + 3 + 9, 1)) *
100 / 256
anag2 = Asc(Mid(trama, ndatosmodbus + ndatosascii + 3 + 10, 1))
* 100 / 256
anag3 = Asc(Mid(trama, ndatosmodbus + ndatosascii + 3 + 11, 1))
* 100 / 256
anag4 = Asc(Mid(trama, ndatosmodbus + ndatosascii + 3 + 12, 1))
* 100 / 256
anag5 = Asc(Mid(trama, ndatosmodbus + ndatosascii + 3 + 13, 1))
* 100 / 256
anag6 = Asc(Mid(trama, ndatosmodbus + ndatosascii + 3 + 14, 1))
* 100 / 256

For i = 8 To 12
    If anag1 = Asc(comp1) Then
        objeto(i).BackColor = &HFF&
    Else
        objeto(i).BackColor = &HFF00&
    End If

    If anag2 = Asc(comp2) Then
        objeto(i).BackColor = &HFF&
    Else
        objeto(i).BackColor = &HFF00&
    End If

    If anag3 = Asc(comp3) Then
        objeto(i).BackColor = &HFF&
    Else
        objeto(i).BackColor = &HFF00&
    End If

    If anag4 = Asc(comp4) Then
        objeto(i).BackColor = &HFF&
    Else
        objeto(i).BackColor = &HFF00&
    End If

    If anag5 = Asc(comp5) Then
        objeto(i).BackColor = &HFF&
    Else

```

```

        objeto(i).BackColor = &HFF00&
    End If
Next i

    Nivel1.Caption = anag1
    Nivel2.Caption = anag2
    Nivel3.Caption = anag3
    Nivel4.Caption = anag4
    Nivel5.Caption = anag5
    Nivel6.Caption = anag6
Else
    MsgBox ("Error de trama de llegada desde el PLC")
End If
textrepcion.Caption = textrepcion.Caption + strData

End Sub

```

Se encarga de salir de la pantalla de operación a principal ver figura 54.

Figura 54. Función salir

```

Private Sub Salir_Click()
    Operación.Hide
    Principal.Show

End Sub

```

Esta se encarga de generar las tramas para el dispositivo, estas son generadas en intervalos de tiempo que fueron especificados por el usuario en la etapa de configuración ver figura 55.

Figura 55. Función Timer

```

Private Sub Timer1_Timer()
    Timer1.Enabled = False

    Select Case tramatipe
        Case 1
            tramamodbus = 4 + direccion + 1 + adinic + npoints
        Case 2
            tramamodbus = 4 + direccion + 2 + adinic + npoints
        Case 3
            tramamodbus = 4 + direccion + 3 + adinic + npoints
        Case Else
    End Select

```



```

tramaenvio = "AAYYY"
sizeascii = trama_ascii.GetLength()
tramaenvio = tramaenvio + sizeascii + tamaño + trama_ascii + tramamodbus +
"ZZ"

If serial = True Then
    Timer1.Enabled = True
    porthmi.Output = tramaenvio
Else
    If ethernet = True Then
        Timer1.Enabled = True
        udpa.SendData tramaenvio
    End If
End If

If objeto(13).BackColor = &HFF& Then
    objeto(13).BackColor = &HFF00&
Else
    objeto(13).BackColor = &HFF&
End If

End Sub

```

- Modulo visor. Este modulo se encarga de estar mirando el estado de otros equipo que se encuentren la red, esta es para saber si están en línea o no.

Esta identificación de estados se hace de forma cíclica con 5 equipos que estén en red, se pregunta el estado de este si no responde, se considerara offline.

En el load se hace una configuración inicial para iniciar el proceso cíclico de identificación ver figura 56.

Figura 56. Load Configuración inicial

```

Private Sub Form_Load()

    mensaje = "HELLO"
    With udp_diagnostics
        .RemoteHost = ipaddress
        .RemotePort = puertoremoto
        .Bind puertolocal
    End With

```

```

udp_diagnostica.SendData mensaje
cuenta = cuenta + 1
End Sub

```

- Botones de cambio de pantalla

Se hacen para pasar de una pantalla a otra

```

Private Sub Opera1_Click()
    Diagnostico.Hide
    Load Operacion
    Operacion.Show
End Sub

```

```

Private Sub Param1_Click()
    Diagnostico.Hide
    Load Parametrizacion
    Parametrizacion.Show
End Sub

```

```

Private Sub Salir1_Click()
    Diagnostico.Hide
    Principal.Show

```

```

End Sub

```

Como sabemos en esta parte lo que se quiere es mirar el estado de los equipos que están en la red, para eso a los equipos se les a programado un código especial el cual solo responderá a una trama udp especifica ver figura 57.

Figura 57. Recolección de datos UDP.

```

Private Sub udp_diagnostica_DataArrival(ByVal bytesTotal As Long)

```

```

    Dim datoudp As String

```

```

    udp_diagnostica.GetData datoudp

```

```

    Select Case cuenta

```

```

        Case 1

```

```

            If Mid(datoudp, 1, 4) = "HERE" Then

```

```

        objeto(0).BackColor = &HFF&
        Label2.Caption = online
    Else
        objeto(0).BackColor = &HFF00&
        Label2.Caption = offline
    End If
    cuenta = cuenta + 1

    With udp_diagnostica
        .RemoteHost = ipaddress1
        .RemotePort = puertoremoto
        .Bind puertolocal
    End With

    udp_diagnostica.SendData "HELLO"
    cuenta = cuenta + 1
Case 2
    If Mid(datoudp, 1, 4) = "HERE" Then
        objeto(1).BackColor = &HFF&
        Label3.Caption = online
    Else
        objeto(1).BackColor = &HFF00&
        Label3.Caption = offline
    End If
    cuenta = cuenta + 1

    With udp_diagnostica
        .RemoteHost = ipaddress2
        .RemotePort = puertoremoto
        .Bind puertolocal
    End With

    udp_diagnostica.SendData "HELLO"
    cuenta = cuenta + 1
Case 3
    If Mid(datoudp, 1, 4) = "HERE" Then
        objeto(2).BackColor = &HFF&
        Label4.Caption = online
    Else
        objeto(2).BackColor = &HFF00&
        Label4.Caption = offline
    End If
    cuenta = cuenta + 1

    With udp_diagnostica
        .RemoteHost = ipaddress2
        .RemotePort = puertoremoto

```

```

        .Bind puertolocal
    End With

    udp_diagnostica.SendData "HELLO"
    cuenta = cuenta + 1
Case 4
    If Mid(datoudp, 1, 4) = "HERE" Then
        objeto(3).BackColor = &HFF&
        Label5.Caption = online
    Else
        objeto(3).BackColor = &HFF00&
        Label5.Caption = offline
    End If
    cuenta = cuenta + 1

    With udp_diagnostica
        .RemoteHost = ipaddress3
        .RemotePort = puertoremoto
        .Bind puertolocal
    End With

    udp_diagnostica.SendData "HELLO"
    cuenta = cuenta + 1
Case 5
    If Mid(datoudp, 1, 4) = "HERE" Then
        objeto(4).BackColor = &HFF&
        Label6.Caption = online
    Else
        objeto(4).BackColor = &HFF00&
        Label6.Caption = offline
    End If
    cuenta = cuenta + 1

    With udp_diagnostica
        .RemoteHost = ipaddress4
        .RemotePort = puertoremoto
        .Bind puertolocal
    End With

    udp_diagnostica.SendData "HELLO"
    cuenta = 1
Case Else
End Select

End Sub

```

3.2. TABLA DE ENTRADAS Y SALIDAS

El sistema se ha definido de la siguiente forma ver figura 58:

Figura 58. Diagrama de entradas y salidas

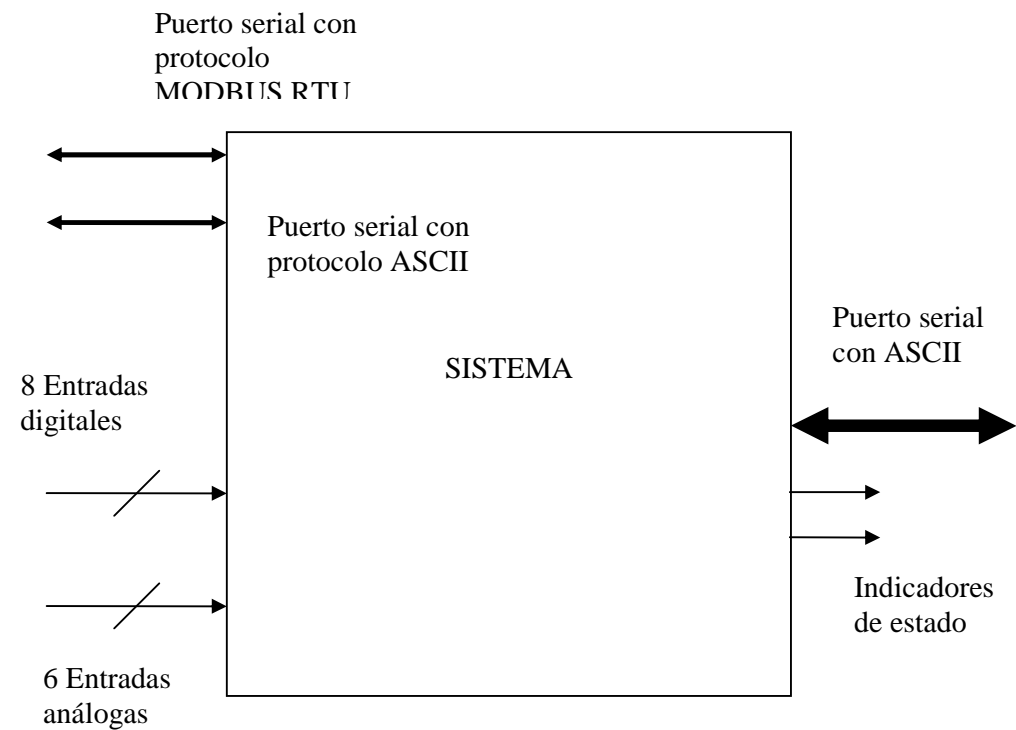
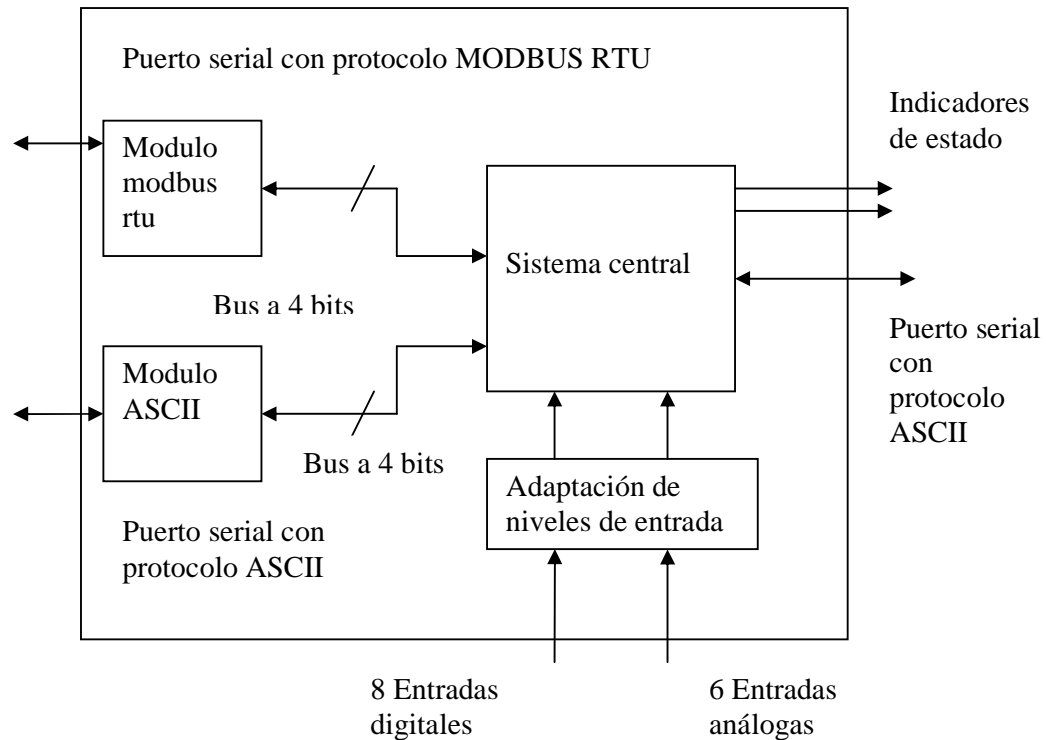


Tabla 16. Entradas y salidas

nombre	tipo	cantidad
análogas	in	6
digitales	in	8
Puertos seriales	Modbus rtu master	3
	ASCII configurable	
	ASCII fijo	

El sistema internamente esta conformado por los siguientes subsistemas ver figura 59:

Figura 59. Subsistemas

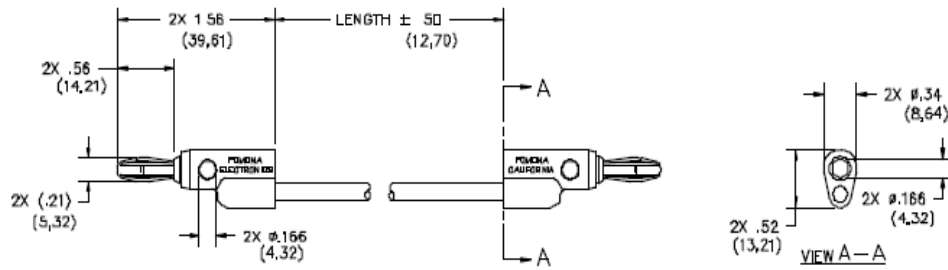


### 3.3. DISEÑO DE CONEXIÓN ENTRE SEÑALES DE INSTRUMENTACIÓN DE CAMPO DE LA TARJETA

Las señales de instrumentación que maneja nuestro sistema son las que son adquiridas por los sistemas analógicos y digitales, para poder obtener estos datos se debe hacer una adecuación de esta, generalmente este tipo de señales vienen ya acondicionadas bajo una norma, la cual no se encuentra en los parámetros del microcontrolador, actualmente existe un estándar aceptado por la mayoría de productos para este tipo de entradas, las analógicas manejan, voltajes de entre 0 v a 5v, en corriente de 4ma a 24ma, las entradas digitales generalmente manejan entre 0v a 24v DC. A partir de esto se diseñó como debería ser la conexión de las señales.

- Conexión de señales a la tarjeta. Como mencionamos antes usaremos el estándar que manejan la mayoría de equipo en la actualidad, y para este tipo de señales se utiliza con mucha frecuencia este tipo de cable, el B-36 Pomona, este tipo de cable es generado por muchas compañías y es de fácil de adquisición, aquí algunas características técnicas ver figura 60.

Figura 60. Características físicas



■ Características eléctricas.

Temperatura de operación: +55C. (+131F) Max.

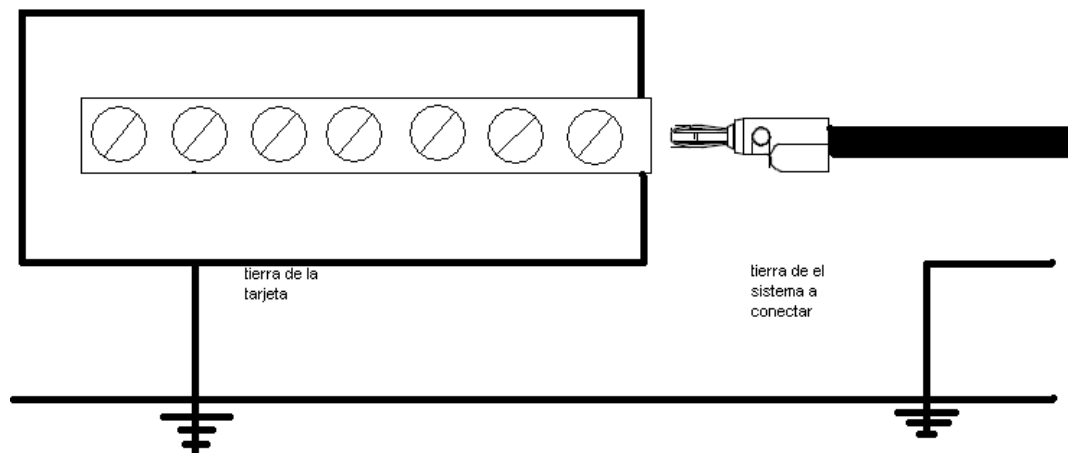
Voltaje de operación: Prueba sin conectar: 30VAC/60VDC Max.

Prueba con voltaje controlado: 5000 WVDC.

Corriente: 15 A.

Este tipo de cable se puede usar para los 2 módulos, análogo y digital, y se conecta de la siguiente manera ver figura 61.

Figura 61. Conexión



### 3.4. DISEÑO DE TARJETA DE COMUNICACIONES Y ADQUISICIÓN DE DATOS

### 3.5. CUANTIFICACIÓN DE OBRAS PARA MONTAJES

Para conectar dos ordenadores en red de área local Ethernet es necesario instalar una tarjeta de red en cada uno de ellos, habilitar la configuración correspondiente y realizar la conexión física con un cable cruzado. Si son más de 2 PC los cables de red local necesarios son normales, y además se necesita un hub o switch. El principal objetivo es tratar de fundamentar la instalación o montaje de una tarjeta o placa de red, cuyo precio de adquisición en el mercado es de 17 o 20 euros lo que equivale a unos 60.000 pesos en el mercado actual. Además, se muestran algunos detalles sobre los cables de red LAN, tanto normales como cruzados. Tanto el cable normal como el cable cruzado se pueden adquirir en una tienda electrónica. Cualquier manipulación sobre el aparato eléctrico o maquinaria exige que previamente se apague y desenchufe, así como especial cuidado en el momento inicial de volver a reiniciar el equipo.

Se pueden usar muchos tipos de cables pero solo se explicara en detalle 2:  
El 10BASE-2 o bien RG58, o BNC o cable coaxial fino, es uno de los cables más clásicos; de un diámetro entorno a 0,5 cm, cada tramo puede tener una longitud máxima de 185 m, con unos 30 ordenadores distribuidos en ese espacio. Es relativamente fácil de usar y montar, aunque resulta algo delicado y puede ser difícil detectar dónde está roto.

El otro es el 10BASE-T o bien UTP, o RJ45 o cable de par trenzado. Externamente es igual que el cable telefónico, incluso en los conectores (las piezas RJ45), aunque no deben confundirse nunca. Con una longitud máxima de unos 100 m por tramo, es muy cómodo de usar, resistente y fácil de diagnosticar errores, aunque necesita usar un aparato denominado hub que encarece la compra. (Para más información ver anexo 1)

El Hub es un elemento que sólo se usa en redes con cables tipo telefónico (10BASE-T, TX, T4...), siendo innecesario en las de cable coaxial. El hub es un elemento de importancia vital, por lo que no conviene discutir en la compra del mismo, especialmente si queremos comprar uno que soporte redes Fast Ethernet. Físicamente todos los hubs son parecidos pequeñas cajas de forma rectangular parecidos a módems externos grandes, con numerosos conectores para los cables y una serie de indicadores luminosos que muestran el estado de la red, lo que resulta fundamental a la hora de diagnosticar problemas. La mayoría de los hubs pueden unirse unos a otros para ampliar la red, aunque para una red del tamaño que nos interesa merece la pena comprar un único hub que gobierne toda la red. Siempre conviene comprar un hub con un par de puertos más de los que se necesitan



### 3.6. PLAN DE ENTRENAMIENTO DE PERSONAL

Se presenta una propuesta de entrenamiento del personal administrativo y técnico de EMCALI para el correcto manejo y operación del dispositivo ha implementar, para tal objetivo se ha desarrollado un plan que involucre capacitación y conocimiento teórico-practico. Este centra sus esfuerzos en herramientas de ofimática, bases de datos e plataformas de interfaz hombre maquina como lo es el Visual Basic (VB).

El plan de formación para el personal de EMCALI tiene como finalidad entrenar al personal técnico que le dará soporte a las unidades administrativas que son la s encargadas de prestar el servicio de telefonía en la ciudad, con lo cual se contará ha corto plazo con un personal calificado en el manejo de herramientas para gestionar sistemas, proyectos y servicios que la empresa demande.

El perfil de los usuarios que podrán participar en el manejo del dispositivo y software de interfaz en visual Basic son los usuarios encargados de manejar herramientas de ofimática para manejar servicios y procesos de la administración y los técnicos de soporte encargados del soporte informático y técnico de cada una de las unidades o dependencias de la empresa.

Se ha estipulado un cronograma de trabajo detallado por módulos con una duración de 8 semanas especificadas en 3 etapas la primera detalla el funcionamiento eléctrico del dispositivo, sus funciones y como esta configurado para el manejo de la información, la segunda detalla como es su instalación su aplicación y como es el montaje en los cuartos de control y por ultimo el manejo de manual de usuario para visualizar la informaron en la herramienta de interfaz (VB). El costo que se ha estipulado para realizar la capacitación es de 900.000 pesos

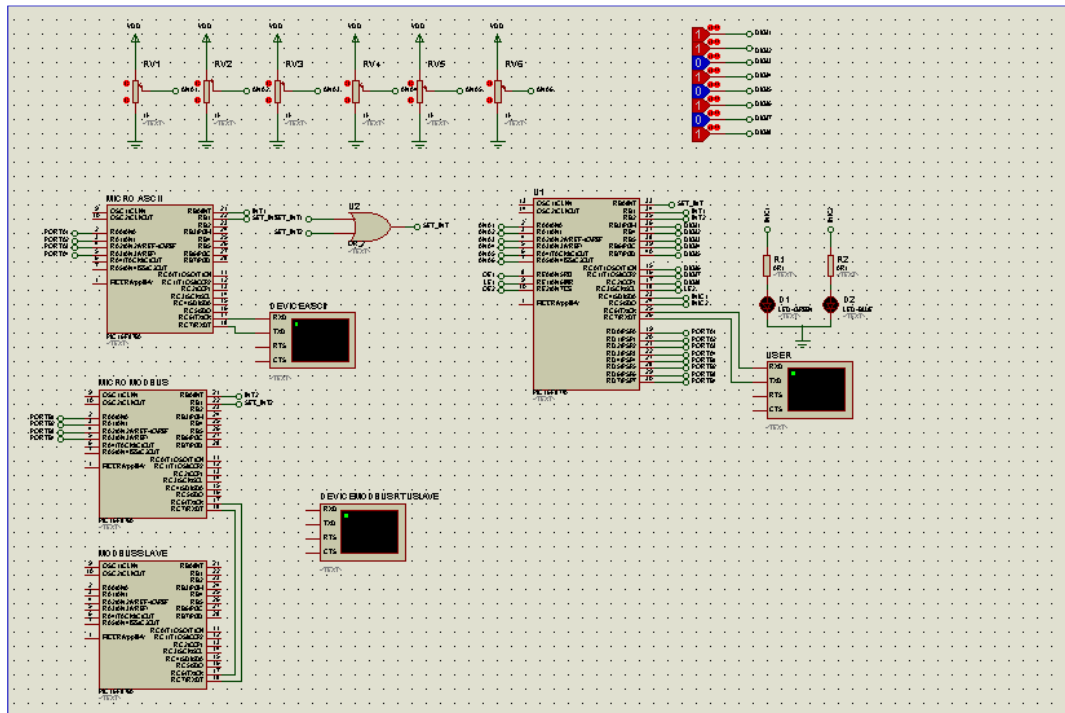
En el anterior capitulo se mostró como cada modulo del sistema es definido y diseñado, además como trabajará con los otros conectados. También se expuso como se haría el montaje del equipo en la empresa y que requisitos tener en cuenta la hora de implementarlo. A continuación se publicará como opera el sistema de forma simulada y a partir de lo mostrado se diseñara un manual de usuario, y lo requerido para un plan de entrenamiento en la empresa, con la respectiva evaluación y pruebas del dispositivo como prototipo final.

#### 4. REALIZAR EVALUACION DISPOSITIVO FINAL

##### 4.1. PRUEBA DEL PROTOTIPO EN LÍNEA CON PC

Vale aclarar que las pruebas se realizaron por medio de simulación mediante el software Proteus, el cual permite hacer este tipos de ejercicios, sin necesidad de montar, claro esta que esto no garantiza que al hacer el montaje físico el dispositivo funcione, el programa solo muestra una simulación en condiciones optimas que nunca se darán.

Figura 62. Simulación híper terminal



En la figura 62 Se muestra la forma del diseño básico en la cual se pueden hacer las pruebas simuladas de las tramas de transmisión y recepción.

La simulación se hace en ese software y no se puede adecuar el software diseñado con Visual BASIC con el simulador Proteus, debido a esto las pruebas se harán individualmente, lo cual se desarrolla con detalle en los puntos siguientes.

## 4.2. PRUEBA Y PUESTA EN MARCHA DE COMUNICACIONES

Las comunicaciones que existen en el desarrollo son las siguientes:

- Comunicaciones del PC
- Comunicaciones del Sistema.

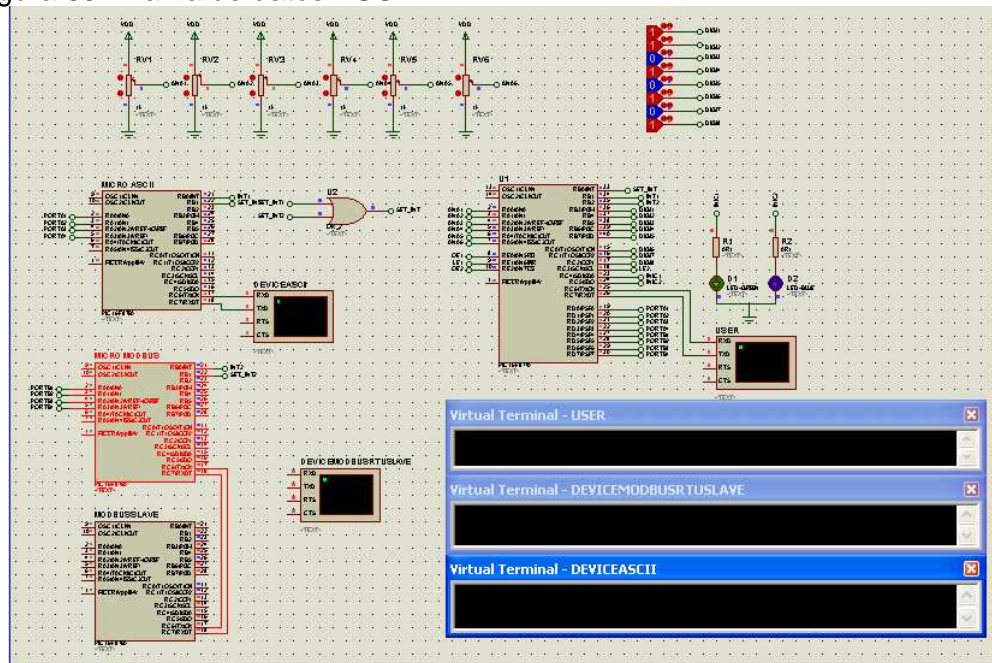
Cada una es un bloque funcional del desarrollo y se utiliza con el propósito único de manejo de información, ya sea entre periféricos, o elementos de red.

4.2.1. Comunicaciones del Sistema. Esta parte del desarrollo fue implementada en el software de monitoreo y el sistema electrónico, explicaremos como se hacen las pruebas del primer caso.

4.2.2. Serial ASCII al PC. Para saber si el software esta corriendo correctamente se miran si las tramas que configuran el dispositivo si esta salen correctamente.

4.2.3. Serial ASCII del Sistema con las otras Comunicaciones. El sistema posee como se sabe 2 puertos seriales por los cuales se maneja información bidireccionalmente por medio de tramas, para hacer las pruebas, se usara Proteus como herramienta de simulación.

Figura 63. Trama de datos ASCII



En la figura 63 se muestra el sistema simulado listo para esperar datos, en el desarrollo se recomendó hacer 2 comunicaciones además de la comunicación entre el sistema y el operador, estas comunicaciones son ASCII y modbus, las ventanas negras representan esas comunicaciones.

Para probar el sistema se usara una trama que generara una respuesta de parte del sistema.

[AAYYY24423812ZZ]

En la figura 64 se muestran los datos de configuración que requiere el sistema para que el ASCII envíe datos y el modbus envíe la trama 01.

Figura 64. Trama de datos ASCII a Modbus rtu

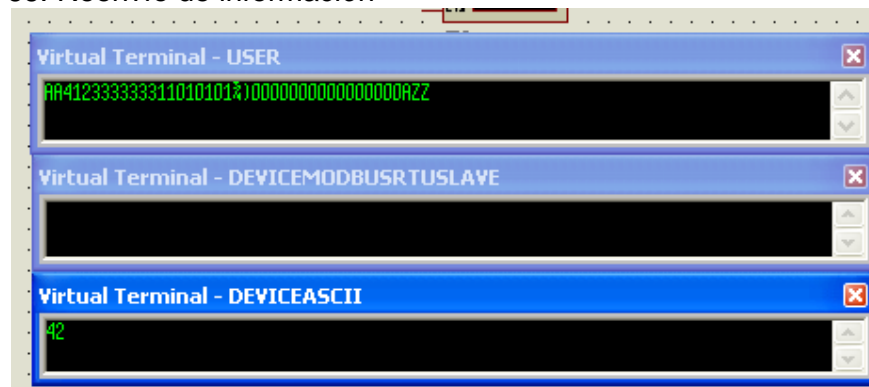


Ya enviada la trama el dispositivo reconoce los campos y separa la información a enviar, en la figura anterior se ve como el terminal ASCII recibe la información de parte del sistema.

Este espera al que el dispositivo ASCII envíe información para luego pasar a la otra red, la Modbus ver figura 65.

The screenshot shows three stacked virtual terminal windows in a Proteus simulation environment. The top window, titled 'Virtual Terminal - USER', is currently empty. The middle window, titled 'Virtual Terminal - DEVICEMODBUSRTUSLAVE', displays the hexadecimal value '00' with a green cursor. The bottom window, titled 'Virtual Terminal - DEVICEASCII', displays the decimal value '42' with a green cursor. Above the windows, a label 'RXD' is visible next to a green indicator light, suggesting a serial communication setup.

Figura 66. Reenvío de información



4.2.4. Comunicaciones del PC. Al PC se le han creado dos módulos de comunicaciones los cuales se activan de acuerdo a lo que el usuario quiere:

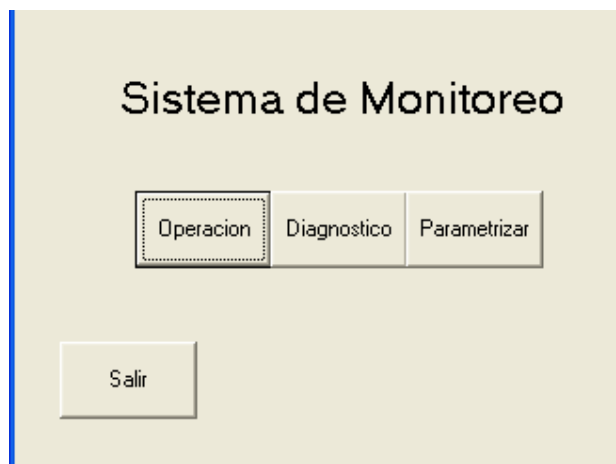
- El modulo Udp/Ethernet
- El modulo ASCII/rs232

La prueba se hace de forma similar en ambos casos, se explicará como se realizó en el caso Udp.

- Conexión Udp.

Al iniciar el programa damos click en Parametrizar ver figura 67

Figura 67. Ventana de interfaz VB



En esta ventana se ingresan los datos de configuración del dispositivo, alarmas etc. (Ver anexo 4 manual de usuario).

Se llenan los campos con lo requerido por los dispositivos a leer, y en la parte de red se selecciona comunicación udp/Ethernet, con esto se dice al software que la lectura del dispositivo se hará por medio de la red, utilizando el puerto Ethernet.

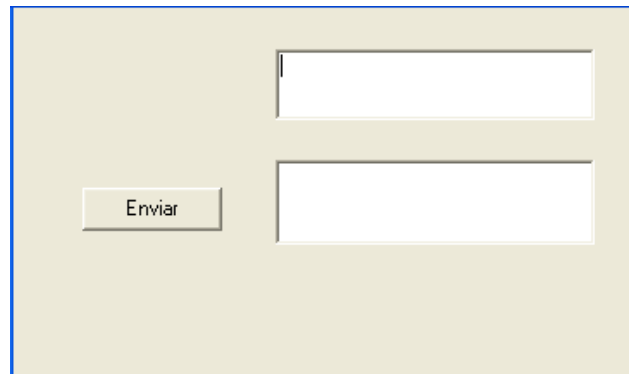
Figura 68. Configuración interfaz VB

Configuracion Modbus RTU	Configuracion de Alarmas de entradas analogicas	Configuracion ASCII
Direccion equipo: <input type="text" value="254"/> Trama Modbus a utilizar: <input type="text" value="Lea estado"/> Desde la direccion 0 hasta: <input type="text" value="4001"/> Leer los puntos desde el 0 hasta: <input type="text" value="4100"/>	Activar cuando la señal pase de: CH1: <input type="text" value="80"/> CH2: <input type="text" value="80"/> CH3: <input type="text" value="90"/> CH4: <input type="text" value="95"/> CH5: <input type="text" value="95"/> En porcentaje	Trama ASCII a enviar: <input type="text" value="aabbzz"/> Configuracion de tiempos de registro: Tiempo de refresco de informacion: <input type="text" value="5"/> Generacion de Archivos cada: <input type="text" value="24"/>
Configuracion de red: Direccion IP del equipo1: <input type="text" value="192.171.1.12"/> Direccion IP del equipo2: <input type="text" value="192.172.2.12"/> Direccion IP del equipo3: <input type="text" value="192.172.1.3.12"/> Direccion IP del equipo4: <input type="text" value="192.172.4.12"/> Direccion IP del equipo5: <input type="text" value="192.172.5.12"/> Puerto al que se conecta: <input type="text" value="8001"/> Puerto local a enlazar: <input type="text" value="8001"/> Comunicacion ethernet/udp: <input checked="" type="radio"/> Comunicacion serial: <input type="radio"/>		
<input type="button" value="Aplicar"/> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/>		

Se da aplicar, luego aceptar, de hay la ventana monitoreo en la cual se verá los efectos de esta configuración ver figura 68.

En la ventana en esta no ocurre nada no hay nada en la red respondiendo a la trama enviada. Como es una prueba del software esta se hace con 2 PC, uno el que posee el software y el otro que muestra los datos enviados por el otro PC, para este propósito se utiliza un código sencillo el cual simplemente muestra lo que llega del otro PC.

Figura 69. Datos de la red

A screenshot of a web interface with a light beige background. It features two empty text input fields stacked vertically on the right side. To the left of these fields is a button labeled 'Enviar'.

Hay se visualizan los datos provenientes de la red figura 69.

Figura 70. Prueba de datos

A screenshot of the same web interface as in Figure 69. The top text input field is now filled with the test data 'AAYYY746aabbzz3|12ZZ'. The 'Enviar' button and the bottom empty text input field remain visible.

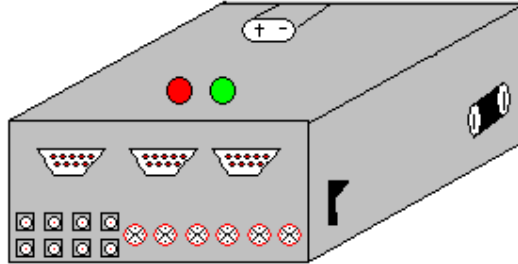
Como se ve los datos provenientes de la red son los correctos, se prueba debido a como llega la trama al PC, con los campos en el orden correcto ver figura 70. ( AAYYY [#ASCII]#MODBUS][DASCII][DMODBUS]ZZ)

4.3. PRUEBA Y PUESTA EN MARCHA DE LAS ESTACIONES DE OPERACIÓN (SOFTWARE VISUAL EN PC Ver anexo 4 manual de usuario.



#### 4.3.1. Generación de planos finales eléctricos ver figuras 71 y 72.

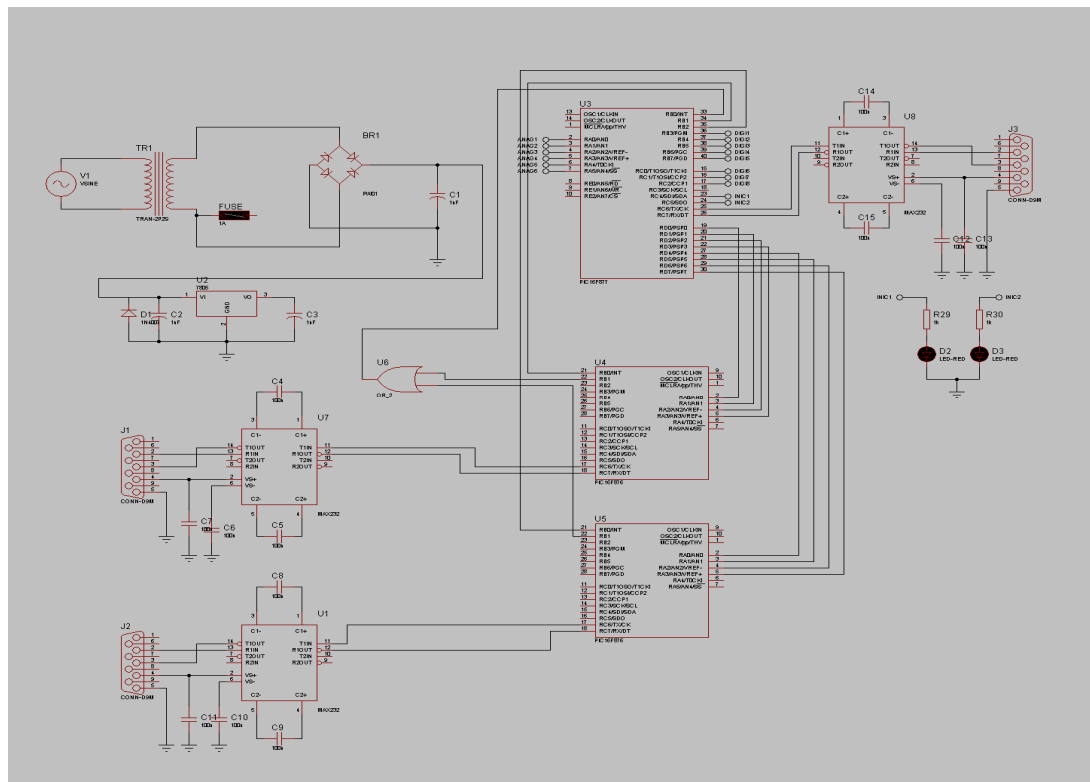
Figura 71. Dispositivo caja negra



#### Características principales

- Alimentación 110 v
- 6 entradas análogas a 10 voltios
- 8 entradas digitales a 24 voltios
- 3 entradas Modbus, ASCII y ASCII a PC
- Fusible de protección
- 2 indicadores Led de funcionamiento y fallas

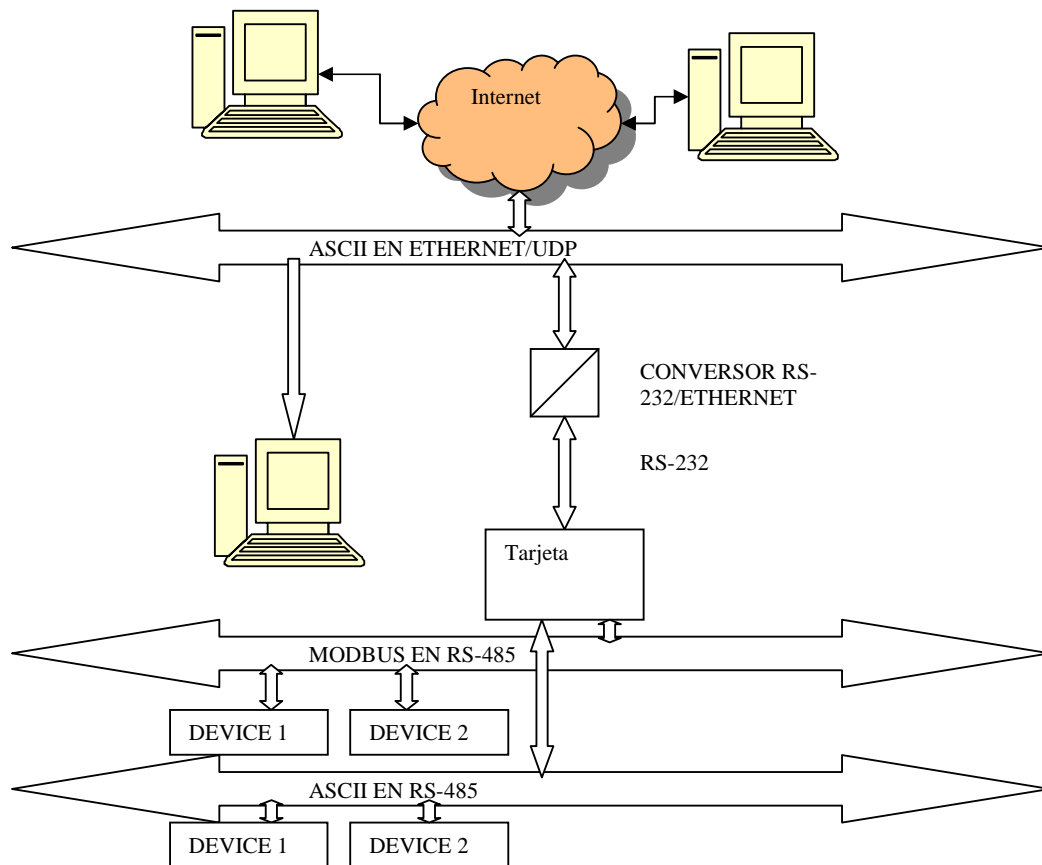
Figura 72. Diagrama eléctrico simulación



#### 4.4. CONFIGURACIÓN, ARQUITECTURA Y COMUNICACIONES SISTEMAS DE CONTROL

El sistema que se ha desarrollado, se ha diseñado con el fin de proporcionar al cliente, la capacidad de monitorear diversas clases de equipo con diferentes métodos, este sistema posee una parte electrónica la cual trabaja junto con un software especializado para poder cumplir este fin. Para esto desde el punto de vista de un sistema de control distribuido (solo de monitoreo), el desarrollo se interpretaría de la siguiente forma.

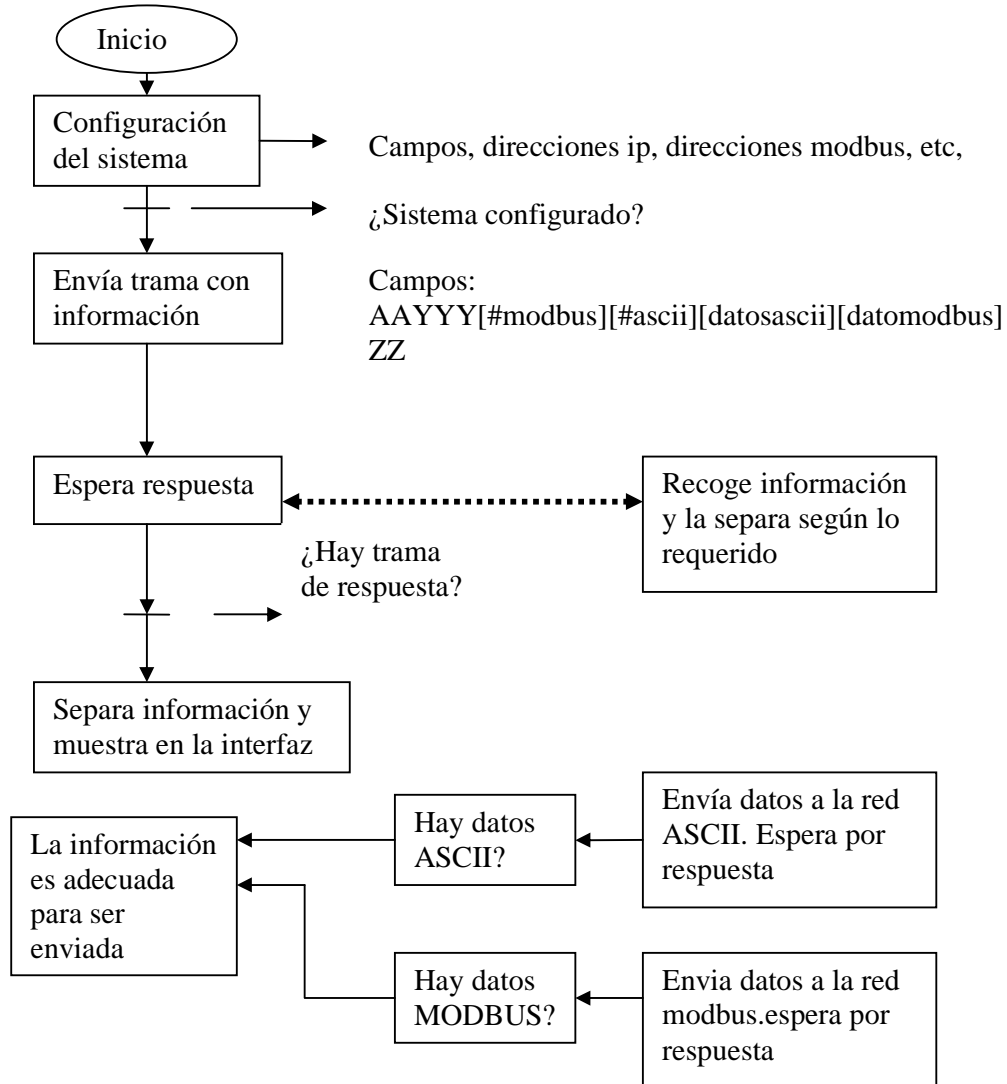
Figura 73. Arquitectura DCS



En la figura 73 se muestra como es la arquitectura del dcs, de esta manera se dimensiona como el sistema se comporta con la red del usuario, y como este debe acoplar este sistema a la misma.

4.4.1. Diagrama del proceso de adquisición de datos. El siguiente diagrama mostrara como el sistema desde el usuario que configura la información deseada, cuando el sistema recibe el pedido y lo ejecuta, hasta que el sistema devuelve al usuario la información pedida ver figura 74.

Figura 74. Diagrama del proceso de adquisición de datos



4.5. ESPECIFICACIONES TÉCNICAS PARA MONTAJES  
Ver anexo 4 (manual de operación y mantenimiento)

## 5. CONCLUSIONES

Se puede concluir que la prioridad en esta etapa preliminar de diseño es que pueda ser monitoreada desde otros puntos, para esto se prestó atención a diferentes posibilidades tecnológicas, equipos, que serán analizados con mas detalle en el siguiente capítulo, también se mostrara esquemas como preámbulo a plantear para la ejecución como un producto.

A partir de lo esbozado en el capítulo anterior, se puede concluir que el sistema trazado posee muchas partes a diseñar, lo cual indica que será importante que el proyecto pueda manejar todos los módulos sin presentar problemas y limitaciones, ya conocido lo anterior se continua con la representación real de los diseños en bloque de cada modulo pretendido, y también se muestra como quedará constituido el sistema en su final.

En el anterior capítulo se mostró como cada modulo del sistema es definido y diseñado, además como trabajará con los otros conectados. También se expuso como se haría el montaje del equipo en la empresa y que requisitos tener en cuenta la hora de implementarlo. A continuación se publicará como opera el sistema de forma simulada y a partir de lo mostrado se diseñara un manual de usuario, y lo requerido para un plan de entrenamiento en la empresa, con la respectiva evaluación y pruebas del dispositivo como prototipo final.

Los resultados más importantes que se espera obtener del proyecto Monitoreo de equipos de apoyo en EMCALI, es la implementación más rápida y económica de sistemas de automatización, por un lado, y una importante mejora en la calidad de los mismos, por el otro. En este sentido, la idea es que EMCALI genere sistemas que incorporen los avances más recientes en el área de optimización de procesos en comunicaciones. Por lo tanto, la calidad de los sistemas generados por EMCALI debiera ser mejor o igual que cualquier sistema que pueda ser generado en forma tradicional y en un tiempo con un costo mucho menor.

Las comunicaciones son una parte importante a la hora de crear dispositivos de monitoreo, ya sea por canales, o para enlazar el sistema con otros sistemas o redes.

Se concluyó por el protocolo MODBUS debido a que este es un protocolo abierto, lo cual implica no tener que pagar por su aplicación, algún chip o

reglamentación que lo establezca. Este protocolo quizá no sea el de mayor difusión en el campo tecnológico, pero es muy obtenido por las compañías a la hora de diseñar sus productos, además es fácil de implementar en sistemas micro controlados o de control distribuido.

El protocolo MODBUS posee varias versiones, de las cuales se enfocó para desarrollar en el proyecto dos, el MODBUS ASCII y el MODBUS RTU. Se escogió este debido a la velocidad en términos de procesar la información y de transmitirla, haciendo diferencia entre los 2 protocolos, la cual consiste en que, el tamaño del dato transmitido serial mente es desigual en ambos casos en términos de capacidad.

La comunicación a cuatro bits bidireccional que se usa para intercambiar información entre microcontroladores posee cierta lentitud, pero se considero poco importante debido a los tiempos de petición que se le dan al sistema. La velocidad de transmisión que se usa en los 2 puertos de monitoreo se considero baja, debido a que se creo el sistema para monitoreo y esto no requiere altas velocidades de transferencia de la información.

Los dispositivos de interfaz entre redes, como el conversor RS232/ETHERNET que se propone en el proyecto, son sistemas que permiten modificar cualquier dispositivo que posea puerto serial en un sistema con acceso a la red Internet.

Se resolvió por EL EIA/TIA-232-E ESTANDAR, debido a que esta muy propagado en el ambiente industrial y es de muy fácil y económica implementación.

El microcontrolador PIC 16f877a y 16f876a son microcontroladores de gama media con una alta disponibilidad de periféricos, lo que da un poder de aplicación importante a la hora de hacer proyectos de gran envergadura

Se prevé que este proyecto permitirá generar tanto nuevos conocimientos como nuevas tecnologías en las áreas de automatización de sistemas productivos y técnicas de modelación y representación de problemas. En particular, para poder lograr los objetivos del proyecto, se debe investigar la mejor forma de formular los modelos y representar los problemas del usuario. También plantea nuevos desafíos, la integración de las tecnologías de automatización y control con otros programas para obtener un sistema computacional amigable para el usuario.

## BIBLIOGRAFIA

Automation & Control Telemecanique: PLC, HMI, Drivers [en línea]. Paris: Schneider Electric, 2005. [Consultado en agosto 20 de 2007]. Disponible en Internet: <http://www.modicon.com/techpubs/toc7.html>

Cisco Systems, Inc. Academia de Networking de Cisco Systems, guía del Segundo año. 3 - 4 ed. Madrid: Pearson Educación S.A., 2004. 810 p.

CREUS, Antonio. Instrumentación industrial: 2 ed. Barcelona: Marcombo Boixareu, 1981. 736 p.

Data Loggres: The Data Loggres [en línea]. Contoocook: MicroDAQ.com, Ltd. 2007. [Consultado 24 de agosto de 2007]. Disponible en Internet: <http://www.microdaq.com/?gclid=CL78 KG574oCFQlaPgodyBOMpQ>.

LOPEZ, Ernesto. Sistema de detección de bloque de movimiento de rotación para el tornillo recolector [CD-ROM]. Santiago de Cali, 2006. 1 CD-ROM. Trabajo de grado (Profesional en Ingeniería Electrónica). Universidad Autónoma de Occidente. Facultad de Ingeniería.

PICmicro MCU Estudio: Que es un microcontrolador? [en línea]. México D.F., 2006 [consultado 12 de julio de 2007]. Disponible en Internet: <http://www.electronicaestudio.com/microcontrolador.htm>.

Profibus: The industrial Communications Community Delivering Greater Enterprise Advantage [en línea]. PI (PROFIBUS & PROFINET Internacional) Karlsruhe: [consultado en septiembre 23 de 2007]. Disponible en Internet <http://www.profibus.com>

QUIJANO ZORRILLA, Alexander; GONZALEZ CORDOBA, Guillermo León. Optimización del monitoreo y control de la información analítica de la planta de acueducto del río cauca [CD-ROM]. Santiago de Cali, 2006. 1 CD-ROM. Trabajo de grado (Profesional en Ingeniería Electrónica). Universidad del Valle. Facultad de Ingeniería.

Electro industria [en línea]. Santiago de Chile: Editora Microbyte Ltda, agosto de 2004. [consultado 16 de agosto de 2007]. Disponible en Internet: <http://www.emb.cl/electroindustria/articulo.mv?xid=32&rank=1>.

3710 ACM: ACM Series Documentación [en línea]. Saanichton: Schneider Electric and Power Measurement, 2006 [Consultado Septiembre 8 de 2007]. Disponible en Internet <http://www.pwrm.com/support/classics/3710ACM/>.

TANENBAUM, Andrew. Redes de computadoras. 4 ed. Madrid: Pearson Educación, 2003. 467 p.

Tutoriales y artículos de fibra optica [en línea]. Mexico D.F: Francisco Ramos, 2008. [Consultado 4 de enero de 2008]. Disponible en Internet: [http://www.radioptica.com/Fibra/material\\_fib.asp](http://www.radioptica.com/Fibra/material_fib.asp)

VALENTINE, Michael. CCNS Exam Cram. 3 ed. Indianapolis: Que Publishing, 2007. 798 p.

VELASQUEZ, Carlos Eduardo. Diseño e implementación de una red para control y monitoreo remoto de una planta de nivel [CD-ROM]. Santiago de Cali, 2003. 1 CD-ROM. Trabajo de grado (Profesional en Ingeniería Electrónica). Universidad Autónoma de Occidente. Facultad de Ingeniería.

¿SWITCHING VS. ROUTING? [en línea]. Caracas: Universidad Central de Venezuela, 2007. [consultado 28 de octubre de 2007]. Disponible en Internet: <http://neutron.ing.ucv.ve/revista-e/No4/articulo.htm>

## ANEXOS

### Anexo A. Tipos de medios de transmisión

- Cable UTP. Una de las primeras decisiones que se enfrenta cuando se planea o se desarrolla un sistema de cableado estructurado es el tipo de medio a utilizar. La norma 568-A reconoce 3 medios diferentes:

#### Cable UTP



La norma TIA 568-A establece las características del cable UTP. La performance del cable y hardware de conexión esta determinado por 3 categorías:

- Categoría 5, hasta 100 Mhz
- Categoría 4, hasta 20 Mhz
- Categoría 3, hasta 16 Mhz.
- Características:
  - Capacidad de alta velocidad
  - Diámetro pequeño
  - Bajo costo
  - Facilidad de instalación
  - Gran base instalada
  - 4 pares sólidos
  - 0.50 mm (24 AWG – diámetro estándar de cable) especificado
  - 0.63 mm (22 AWG – diámetro estándar de cable) permitid
- Ventajas. El UTP es capaz de soportar velocidades de datos hasta de 155 Mbps o más. Con el uso de conectores IDC con el cable UTP, se ha reducido el tiempo de instalación. Por su menor diámetro permite una mayor flexibilidad, radio de curvatura menor.



- Cable STP. (par trenzado blindado) de ¼, en cable de cobre de 100 Ohms y en cobre de 150 Ohms-2 pares, calibre 22.

La norma 568-A ahora acepta STP-A, la cual extiende estos sistemas para tasas de hasta 300 Mhz. De hecho este sistema puede llevar una señal Token Ring de 16 Mbps y una imagen de banda ancha a 550 Mhz simultáneamente.

- Características:
  - 0.63 mm (22 AWG) diámetro estándar de cable
  - 150 Ohms de impedancia característica
- Ventajas
  - Limita el escape de EMI (interferencia electromagnética) del cable al ambiente
  - Limita el ingreso de ruido exterior a los conductores
  - Principalmente exigido por normas europeas
- Desventajas
  - Costo más elevado que el UTP
  - Más dificultad para instalar
  - Los canales individuales aun sufren dentro de un apantallamiento de diafonía entre uno y otro.

- Fibra óptica

- Características de los sistemas en fibra óptica.

- La velocidad de transmisión más rápida, ya sea en datos, voz y video
  - El más bajo porcentaje de error de transmisión
  - Total inmunidad a interferencia de electromagnetismo o de radio frecuencia
  - Resistencia a la corrosión, fuego y químicos
  - Extremadamente pequeño en peso y tamaño
  - Las ventajas que ofrece la fibra óptica autofinancian su costo en comparación de las otras posibilidades.
  - Ventajas
    - Mayor capacidad debido al ancho de banda mayor disponible en las frecuencias ópticas
    - Pueden transmitir a velocidades mucho mas altas de lo que emisores y transmisores actuales lo permiten, son estos los que limitan la velocidad de transmisión
    - Gran flexibilidad, el radio de curvatura puede ser inferior a 1 cm, lo que facilita su instalación

- Gran ligereza, el peso es del orden de algunos gramos por kilómetro, lo que resulta unas nueve veces menos que el de un cable convencional
- Inmunidad total a las perturbaciones de origen electromagnético, lo que implica una calidad de transmisor excelente en aplicaciones que requieren alto nivel de confidencialidad
- Inmunidad a interferencia estática debida a las fuentes de ruido
- Desventajas
- Perdidas en los cables por absorción de luz, la cual puede ser convertida en calor
- Perdidas por dispersión de Rayleigh o materiales: luz directa que se escapa por la cubierta al chocar contra una irregularidad del vidrio en el proceso de fabricación
- Perdidas de radiación: causada por dobleces e irregularidades en la fibra
- Perdida modal: causada por diferencias de tiempos de propagación de los rayos de luz que toman diferentes trayectorias por una fibra
- Perdidas de acoplamiento: en las conexiones de fuentes de fibra, fibra a fibra y/o fibra a foto detector, que son causadas por problemas de alineación.

### Fibra óptica



Parámetros eléctricos que se miden:

- Atenuación en función de la frecuencia (dB)
- Impedancia característica del cable (Ohms)
- Acoplamiento del punto más cercano (NEXT-dB)
- Relación entre atenuación y Crosstalk (ACR-dB)
- Capacitancia (pf/m)
- Resistencia en DC /Ohms/m)
- Velocidad de propagación nominal

- Área de trabajo. Sus componentes llevan la comunicación desde la unión de la toma o salida y su conector donde termina el sistema de cableado horizontal, al equipo o estación de trabajo del usuario.

#### Área de trabajo



- Terminaciones mecánicas de los cables y las conexiones
- Roseta: Es un dispositivo de salida, el cual esta conformado por una roseta de polietileno y un jack RJ-45 para datos o RJ-11 para telefonía, este dispositivo se encuentra en la terminación del cable horizontal.

Jack

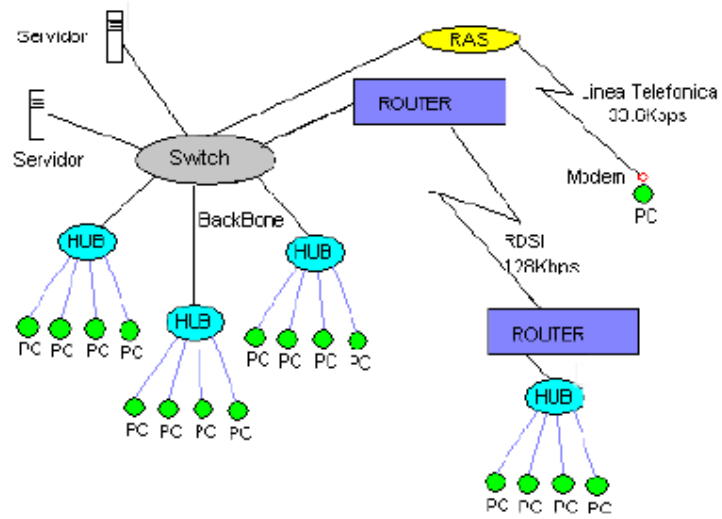


#### Capuchas para RJ-45



- Especificaciones de instalación. Primero es tender el cable de los habites de telecomunicaciones hasta la ubicación de cada toma o salida. La máxima extensión de estiramiento para categoría 5, es de 25 libras por pie. Mayor tensión puede distender el trenzado, en efecto, destrenzar los pares.

## Arquitectura del cableado



## Anexo B. Selección de normas a seguir

### Norma isa-s5.1

#### Generalidades

A) Cada instrumento debe identificarse con sistema de letras que lo clasifique funcionalmente.

B) El número de letras funcionales para un instrumento debe ser mínimo, no excediendo de cuatro. Para ello conviene:

1. Disponer las letras en subgrupos. Por ejemplo, un transmisor registrador de relación de caudales con un interruptor de alarma de relación de caudales puede identificarse con dos círculos uno con FFRT-3 y el otro FFS-3.
2. En un instrumento que indica y registra la misma variable medida puede omitirse la letra I (indicación).
3. Los bucles de instrumentos de un proyecto o secciones de un proyecto deben identificarse con una secuencia única de números. Esta puede empezar con el número 1 o cualquier otro número conveniente, tal como 301 o 1201 que puede incorporar información codificada tal como área de planta.
4. Si un bucle dado tiene más de un instrumento con la misma identificación funcional, es preferible añadir un sufijo, ejemplo FV-2A, FV-2B, FV-2C, etc., o TE-25-1, TE-25-2, TE-25-3, etc. Estos sufijos pueden añadirse obedeciendo a las siguientes reglas:

a) Deben emplearse letras mayúsculas, A, B, C, etc.

b) En un instrumento tal como un registrador de temperatura multipunto que imprime números para identificación de los puntos, los elementos primarios pueden numerarse TE-25-1, TE-25-2, TE-25-3, etc.

c) Las subdivisiones interiores de un bucle pueden designarse por sufijos formados por letras y números.

d) Un instrumento que realiza dos o más funciones puede designarse por todas sus funciones. Por ejemplo, un registrador de caudal FR-2 con pluma de presión PR-4 se designa preferentemente FR-2/PR-4 o bien UR-7; un registrador de presión de dos plumas como PR-7/8; y una ventanilla de alarma para temperatura alta y baja como TAH/L-9.

e) Los accesorios para instrumentos tales como rotámetros de purga, filtros mano reductores y pots de sellos que no están representados explícitamente en un diagrama de flujo, pero que necesitan una identificación para otros usos deben tenerla de acuerdo con su función y deben emplear el mismo número del bucle que el del instrumento asociado. Alternativamente, los accesorios pueden emplear el mismo número de identificación que el de sus instrumentos asociados, pero con palabras aclaratorias si ello es necesario.

Por consiguiente, una brida para una placa-orificio FE-7 debe designarse como FX-7 o bien como FE-7 brida. Un rotámetro regulador de purga asociado con un manómetro PI-8 debe identificarse como FICV-8, pero puede también marcarse PI-8 purga. Una sonda empleada con un termómetro TI-9 será TW-9, o bien, TI-9 sonda.

- Para cubrir las designaciones no normalizadas que pueden emplearse repetidamente en un proyecto se han previsto letras libres. Estas letras pueden tener un significado como primera letra y otro, como letra sucesiva. Por ejemplo, la letra N puede representar como primera letra el módulo de elasticidad y como sucesiva un osciloscopio.
- La letra sin clasificar X, puede emplearse en las designaciones no indicadas que se utilicen sólo una vez o un número limitado de veces. Se recomienda que su significado figure en el exterior del círculo de identificación del instrumento. Ejemplo: XR-3 registrador de vibración.
- Cualquier letra primera si se utiliza con las letras de modificación D (diferencial), F (relación) o Q (integración) o cualquier combinación de las mismas cambia su significado para representar una nueva variable medida. Por ejemplo, los instrumentos TDI y TI miden dos variables distintas, la temperatura diferencial y la temperatura, respectivamente.
- La letra A para análisis, abarca todos los análisis no indicados en la tabla que no están cubiertos por una letra libre. Es conveniente definir el tipo de análisis al lado del símbolo en el diagrama de proceso.
- El empleo de la letra U como multivariable en lugar de una combinación de primeras letras, es opcional.
- El empleo de los términos de modificaciones alto, medio, bajo, medio o intermedio y exploración, es preferible pero opcional.
- El término seguridad, debe aplicarse sólo a elementos primarios y a elementos finales de control que protejan contra condiciones de emergencia (peligrosas para el equipo o el personal). Por esto motiva, una válvula autorreguladora de presión que regula la presión de salida de un sistema, mediante el alivio o escape de fluido al exterior, debe ser

PCV, pero si esta misma válvula se emplea contra condiciones de emergencia, se designa PSV.

- La designación PSV se aplica a todas las válvulas proyectadas para proteger contra condiciones de emergencia de presión sin tener en cuenta si las características de la válvula y la forma de trabajo la colocan en la categoría de válvula de seguridad, válvula de alivio, o válvula de seguridad de alivio.
- La letra de función pasiva vidrio, se aplica a los instrumentos que proporcionan una visión directa no calibrada del proceso.
- La letra indicación se refiere a la lectura de una medida real de proceso. No se aplica a la escala de ajuste manual de la variable si no hay indicación de ésta.
- Una luz piloto que es parte de un bucle de control debe designarse por una primera letra seguida de la letra sucesiva L. Por ejemplo, una luz piloto que indica un período de tiempo terminado se designará KL. Sin embargo, si se desea identificar una luz piloto fuera del bucle de control, la luz piloto puede designarse en la misma forma o bien alternativamente por una letra única L. Por ejemplo, una luz piloto de marcha de un motor eléctrico puede identificarse EL, suponiendo que la variable medida adecuada es la tensión, o bien XL, suponiendo que la luz es excitada por los contactos eléctricos auxiliares del arrancador del motor, o bien simplemente L. La actuación de la luz piloto puede ser acompañada por una señal audible.
- El empleo de la letra U como multifunción en lugar de una combinación de otras letras, es opcional.
- Se supone que las funciones asociadas con el uso de la letra sucesiva Y se definirán en el exterior del símbolo del instrumento cuando sea conveniente hacerlo así.
- Los términos alto, bajo y medio o intermedio deben corresponder a valores de la variable medida, no a los de la señal a menos que se indique de otro modo. Por ejemplo, una alarma de nivel alto derivada de una señal de un transmisor de nivel de acción inversa debe designarse LAH incluso aunque la alarma sea actuada cuando la señal cae a un valor bajo.
- Los términos alto y bajo, cuando se aplican a válvulas, o a otros dispositivos de cierre- apertura, se definen como sigue:
- Alto: indica que la válvula está, o se aproxima a la posición de apertura completa.

Bajo: denota que se acerca o está en la posición completamente cerrada.  
Figuran a continuación los símbolos a emplear en los planos y dibujos de representación de instrumentos en los procesos industriales.

Se sugieren las siguientes abreviaturas para representar el tipo de alimentación (o bien de purga de fluidos)

1. AB Alimentación de aire
2. ES Alimentación eléctrica
3. GS Alimentación de gas
4. HS Alimentación hidráulica
5. NS Alimentación de nitrógeno
6. SS Alimentación de vapor
7. WS Alimentación de agua

Norma ISA-S5.3. El objeto de esta norma es documentar los instrumentos formados por ordenadores, controladores programables, mini ordenadores y sistemas a microprocesador que disponen de control compartido, visualización compartida y otras características de interfase. Los símbolos representan la interfase con los equipos anteriores de la instrumentación de campo, de la instrumentación de la sala de control y de otros tipos de hardware.

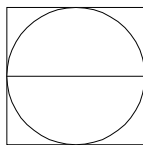
El tamaño de los símbolos debe ser conforme a la norma ISA-S5.1, a la que complementa.

Símbolos de visualización del control distribuido/compartido

Accesible normalmente al operador-indicador/controlador/registrador o punto de alarma.

- Visualización compartida.
- Visualización y control compartidos.
- Acceso limitado a la red de comunicaciones.
- Interfase del operador en la red de comunicaciones.

Accesible al operador

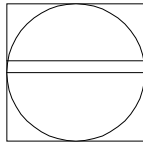


Dispositivo de interfase auxiliar del operador.

- Montado en panel; carátula analógica; no está montado normalmente en la consola principal del operador.
- Controlador de reserva o estación manual.
- El acceso puede estar limitado a la red de comunicaciones.
- Interfase del operador vía la red de comunicaciones.



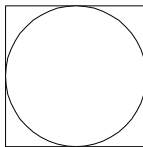
Interfase auxiliar



### 3. No accesible normalmente al operador

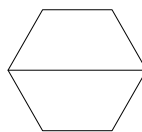
- Controlador ciego compartido.
- Visualización compartida instalada en campo
- Cálculo, acondicionamiento de señal en controlador compartido.
- Puede estar en la red de comunicaciones.
- Normalmente operación ciega
- Puede ser alterado por la configuración.

No accesible normalmente al operador



- Símbolos del ordenador. A utilizar cuando los sistemas incluyen componentes identificados como ordenadores, diferentes de un procesador integral que excita las varias funciones de un sistema de control distribuido. El componente ordenador puede ser integrado en el sistema, vía la red de datos, o puede ser un ordenador aislado. Normalmente accesible al operador-indicador/controlador/registrador o punto de alarma. Utilizado usualmente para indicar la pantalla de vídeo.

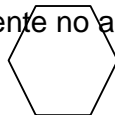
Normalmente accesible



Normalmente no accesible al operador.

- Interfase entrada/salida.
- Cálculo /acondicionamiento de señal dentro de un ordenador
- Puede usarse como un controlador ciego o como módulo de cálculo de software

Normalmente no accesible

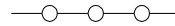


- SÍMBOLOS COMUNES

- Red del sistema.

1. Usado para indicar una red de software, o conexiones entre funciones suministradas en el sistema del fabricante.
2. Alternativamente, la red puede ser mostrada implícitamente por símbolos contiguos.

Conexión de software o datos



3. Puede utilizarse para indicar una red de comunicaciones a opción del usuario.

#### Registradores y otros sistemas de retención de datos históricos

Los registradores convencionales, tales como los de gráfico de banda se mostrarán de acuerdo con ISA-S51


En los registradores asignables utilice el símbolo 1.

El almacenamiento en masa de largo plazo de una variable de proceso mediante memorias digitales como cinta, disco, etc., debe representarse de acuerdo con los símbolos de visualización de control distribuido/compartido o símbolos de ordenador de esta norma, dependiendo de la localización del aparato.

- Identificación. Los códigos de identificación de esta norma deben cumplir con ISA-S5.1 con las siguientes adiciones.
- Alarmas de software. Las alarmas de software pueden ser identificadas situando letras de designación de la tabla 1.1 de ISA-S5.1 en las líneas de señal de entrada o de salida de los controladores, o de otro componente específico integral del sistema. Ver Alarmas que aparecen posteriormente.

## Anexo C. Hoja de especificaciones

### Microcontrolador PIC16F87X



# PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

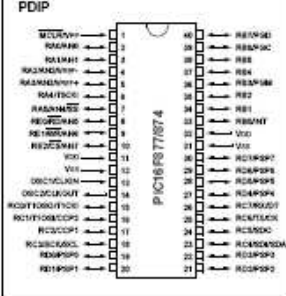
**Devices included in this Data Sheet:**

- PIC16F873
- PIC16F876
- PIC16F874
- PIC16F877

**Microcontroller Core Features:**

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
- DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory
- Up to 384 x 8 bytes of Data Memory (RAM)
- Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature ranges
- Low-power consumption:
  - < 0.5 mA typical @ 3V, 4 MHz
  - 20 µA typical @ 3V, 32 kHz
  - < 1 µA typical standby current

**Pin Diagram**



**Peripheral Features:**

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I2C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

© 2001 Microchip Technology Inc.

0530292C-page 1

Fuente: [www.data.sheet.com/pic16f87x](http://www.data.sheet.com/pic16f87x)

Anexo D. Manual de usuario y mantenimiento

Manual de operaciones del dispositivo de monitoreo de señales y  
comunicaciones  
Diseñado para la empresa EMCALI E.S.P.

Diseñado por  
Daniel Felipe Caicedo  
Cristian Felipe Bravo

UNIVERSIDAD AUTONOMA DE OCCIDENTE  
FACULTAD DE INGENIERIA  
DEPARTAMENTO DE INGENIERIA ELECTRONICA  
PROGRAMA DE INGENIERIA ELECTRONICA  
SANTIAGO DE CALI  
2007

## CONTENIDO

	Pág.
1. Generalidades.....	3
2. Características.....	3
2.1. Características eléctricas y físicas.....	3
2.2. Características de red.....	3
2.3. Configuraciones.....	4
2.3.1. Configuración red Modbus.....	4
2.3.2. Configuración de red ASCII.....	4
2.4. Dispositivos que le son compatibles.....	4
3 Software de monitoreo.....	5
3.1 Configuración.....	5
Glosario.....	9

### 1. GENERALIDADES.

El dispositivo se creo con el objetivo de mejorar el desempeño de la empresa EMCALI en la sección de mantenimiento de equipos. Esta empresa posee una carente capacidad de control de los dispositivos que sufren daños, esto a llevado a la empresa a tener que hacer gastos innecesarios que se reflejan en una pobre e ineficiente uso de los recursos del gobierno, de hay la necesidad de ellos de poder obtener algo que les pueda ayudar el manejo de estos eventos. El dispositivo que ellos pretendían, tenia que poseer la capacidad de manejar la información y enviarla a un PC central, y también la capacidad de poder ser observado desde otro PC diferente, con esta características el dispositivo fue creado tratando de hacerlo de la forma mas simple posible. A continuación se mostrara el producto diseñado para la empresa que cumple con esas características.

### 2. CARACTERÍSTICAS

El dispositivo posee unas características que deben ser tenidas en cuenta a la hora de montarlo en cualquier red.

#### 2.1. Características eléctricas y físicas.

- 2 puertos seriales RS -232, compatibles con Redes RS-485 para manejo de más dispositivos.
- Un puerto compatible con conversor RS-232/ETHERNET.
- Maneja comunicaciones en cada puerto, en 2 maneja ASCII configurable, Velocidad de transmisión de 2400 bauds, 1 bit start , 7 bit de datos, 1 stop, no paridad, en otro maneja Modbus rtu Maestro a 2400 bauds, 1 bit start, 8 bits datos, 2 stop, no paridad.
- 8 entradas digitales acondicionadas de 0 a 24 Vdc.
- 6 entradas análogas acondicionadas de 0 a 10 V dc.
- Señales de diagnósticos (tanto físicas, como implementadas en el sistema).

## 2.2. CARACTERÍSTICAS DE RED.

El dispositivo posee la capacidad de poder hacer parte de una red, y de manejar unas subredes, para su buen desempeño hay que tener en cuenta las siguientes condiciones a la hora de implementarlo.

- Si se desea convertir el puerto Modbus en una red se debe saber que al dispositivo solo se le pueden conectar 5 equipos en la subred, esto debido a la capacidad del dispositivo para procesar información, lo mismo pasa con la subred ASCII.
- El punto de conexión del dispositivo a la el PC central es también para conectar a un conversor de RS-232/ETHERNET, para esto se debe tener en cuenta que este dispositivo debe cumplir con las características eléctricas requeridas por el sistema de monitoreo, una de esas es la velocidad de transmisión de datos, en algunos de estos conversores esta es configurable, pero en otros no.
- El lado Ethernet debe tener montado un protocolo de intercambio de información, en este caso UDP, esto debido a que el software que se diseña utiliza este método de intercambio de datos.

## 2.3. CONFIGURACIONES

El sistema se puede configurar sus tramas para cada una de sus sub. Redes, estas configuraciones se hacen por el operador desde el PC central (sea el caso). La información es enviada al dispositivo que según lo requerido genera las diferentes tramas para cada red.

2.3.1. Configuración de la red Modbus. El protocolo Modbus es un protocolo abierto creado por la empresa Modicon, para generar un estándar de comunicaciones para sus productos, el propósito de este manual no es explicar el protocolo Modbus, por esto recomendamos al lector leer los

anexos de este documento provenientes de la norma Modbus RTU Master, en donde se muestran la tramas implementadas y ejemplos.

2.3.2. Configuración de la red ASCII. Como su nombre lo indica el protocolo ASCII, maneja caracteres ASCII para el reconocimiento de las tramas provenientes de los dispositivos, algunos dispositivos manejan protocolos genéricos de este tipo que el ingeniero a cargo o el operario puede manejar sin mayor dificultad, de igual forma se mostrara un ejemplo de una implementación par a su mayor entendimiento.( la configuración del dispositivo desde el software se ara con mayor profundidad)

Suponga que posee 2 dispositivos ASCII, el primero reconoce tramas del tipo

AA ["información"] ZZ y responde de igual forma.

El segundo dispositivo actúa de forma similar:

AB ["información"] ZZ y responde de igual forma.

Estas serian las tramas que usted ingresaría por el software para obtener datos de cada dispositivo en la red.

#### 2.4. Dispositivos que le son compatibles.

Como se había especificado el dispositivo traba junto con otro para poder así tener acceso a la información que este suministra, estos dispositivos son lo conversores RS-232/ETHERNET de los cuales se encuentra información muy fácilmente, pero para una mejor calidad de l dispositivo diseñado se recomienda mirar los anexos en este documento, en los cuales mostramos algunos de estos sistemas que consideramos trabajaran con el equipo.

#### 3. Software de monitoreo.

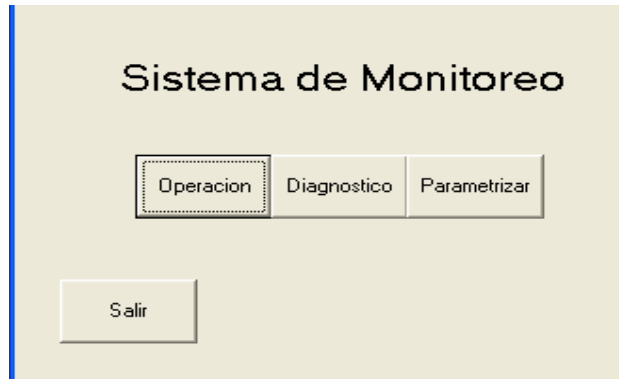
Para la empresa EMCALI E.S.P. se ha creado un software de prueba del sistema, este no hacia parte del diseño, este esta limitado pero se le harán cambios de acuerdo a lo que requiera la empresa, este software esta configurado de la siguiente manera.

- Visor de estado de las tramas entrante (tanto ASCII como Modbus)
- Visor de estado de las entradas digitales.
- Visor y generador de alarmas de las entradas analógicas.
- Panel de configuración del sistema.
- Panel de diagnostico de los sistemas que se encuentran disponibles en la red.
- Capacidad de manejar al dispositivo por medio del canal serial o el canal Ethernet.

### 3.1. Configuración.

A continuación se mostrara un ejemplo de configuración del dispositivo.  
Al iniciar el programa damos click en Parametrizar ver figura 1.

Fig. 1. Pantalla inicial



En esta ventana se ingresan los datos de configuración del dispositivo, alarmas etc. (se explica con más detalle en el manual de usuario).

Se llenan los campos con lo requerido por los dispositivos a leer, y en la parte de red se selecciona comunicación udp/Ethernet, con esto le decimos al software que la lectura del dispositivo se hará por medio de la red, utilizando el puerto Ethernet ver figura 2.



Figura 2. Configuraciones

Configuracion Modbus RTU	Configuracion de Alarmas de entradas analogicas	Configuracion ASCII
Direccion equipo: <input type="text" value="254"/>	Activar cuando la señal pase de:	Trama ASCII a enviar: <input type="text" value="aabbzz"/>
Trama Modbus a utilizar: <input type="text" value="Lea estado"/>	CH1: <input type="text" value="80"/> CH2: <input type="text" value="80"/> CH3: <input type="text" value="90"/> CH4: <input type="text" value="95"/> CH5: <input type="text" value="95"/>	Configuracion de tiempos de registro
Desde la direccion 0 hasta: <input type="text" value="4001"/>	En porcentaje	Tiempo de refresco de informacion: <input type="text" value="5"/>
Leer los puntos desde el 0 hasta: <input type="text" value="4100"/>		Generacion de Archivos cada: <input type="text" value="24"/>
Configuracion de red		
Direccion IP del equipo1: <input type="text" value="192.171.1.12"/>	Puerto al que se conecta: <input type="text" value="8001"/>	
Direccion IP del equipo2: <input type="text" value="192.172.2.12"/>	Puerto local a enlazar: <input type="text" value="8001"/>	
Direccion IP del equipo3: <input type="text" value="192.172.1.3.12"/>	Comunicacion ethernet/udp: <input checked="" type="radio"/>	
Direccion IP del equipo4: <input type="text" value="192.172.4.12"/>	Comunicacion serial: <input type="radio"/>	
Direccion IP del equipo5: <input type="text" value="192.172.5.12"/>		
<input type="button" value="Aplicar"/> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/>		

Se da aplicar Luego aceptar de hay vamos a la ventana monitoreo en la cual se vera los efectos de esta configuración ver figura 3.

Figura 3. Alarmas generadas

Alarmas entradas digitales		Alarmas de entradas analogas		Datos Modbus	Menu
			Niveles(%)		
	Alarma1	CH0	0	<input type="text"/>	<input type="button" value="Diagnostico"/>
	Alarma2	CH1	0		<input type="button" value="Parametrizacion"/>
	Alarma3	CH2	0		<input type="button" value="Salir"/>
	Alarma4	CH3	0		
	Alarma5	CH4	0	<input type="text"/>	
	Alarma6				
	Alarma7				
	Alarma8				

Trama de llegada

También se podrá ver si hay otros dispositivos en la red y cual es su estado ver figura 4.

Figura.4. visor de estado de los equipos

Sistemas en linea		Menu
Sistemas	Estado	
		<input type="button" value="Operacion"/>
		<input type="button" value="Parametrizacion"/>
		<input type="button" value="Principal"/>
		<input type="button" value="Salir"/>